



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

**POSOUZENÍ INFORMAČNÍHO SYSTÉMU FIRMY A NÁVRH
ZMĚN**

INFORMATION SYSTEM ASSESSMENT AND PROPOSAL FOR ICT MODIFICATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Hlaváč

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Miloš Koch, CSc.

BRNO 2017

Zadání diplomové práce

Ústav: Ústav informatiky
Student: **Bc. Petr Hlaváč**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Informační management
Vedoucí práce: **doc. Ing. Miloš Koch, CSc.**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Posouzení informačního systému firmy a návrh změn

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza problému
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Analyzovat stávající stav informačního systému vybrané organizace a jeho efektivnosti, posoudit tento stav a navrhnout změny, směřující ke zlepšení stávajícího stavu a eliminaci nalezených rizik.

Základní literární prameny:

BASL, Josef a Roman BLAŽIČEK. Podnikové informační systémy: podnik v informační společnosti. 3. aktualiz. a dopl. vyd. Praha: Grada, 2012. 323 s. ISBN 978-80-247-4307-3.

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2. přeprac. a aktualiz. vyd. Praha: Grada. 2009, 496 s. ISBN 978-80-247-2615-1.

MOLNÁR, Zdeněk. Efektivnost informačních systémů. 2. rozš. vyd. Praha: Ikar, 2000. 178 s. ISBN 80-247-0087-5.

SCHWALBE, Kathy. Řízení projektů v IT. Brno: Computer Press, 2007. 720 s. ISBN 978-80-251-1526-8.

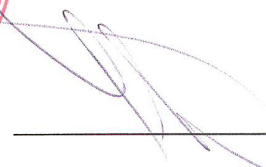
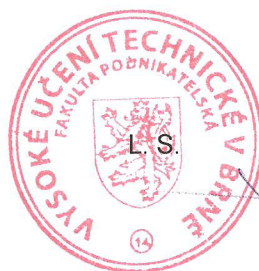
SODOMKA, Petr a Hana KLČOVÁ. Informační systémy v podnikové praxi. 2. aktualiz. a rozš. vyd. Brno: Computer Press, 2010. 501 s. ISBN 978-80-251-2878-7.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2016/17.

V Brně, dne 28. 2. 2017



doc. RNDr. Bedřich Půža, CSc.
ředitel



doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

ABSTRAKT

Diplomová práce se zabývá posouzením informačního výrobního systému PHARIS a následným návrhem změn pro efektivnější práci s tímto informačním systémem. Diplomová práce je zaměřena na výrobní informační systém (MES). Hlavním cílem diplomové práce bude návrh opatření a změn pro optimalizaci tohoto informačního systému. Po zavedení těchto změn by se měla eliminovat slabá místa systému a přinést zjednodušení práce se systémem.

ABSTRACT

This thesis deals with the assessment of the information production system PHARIS and subsequent changes for more efficient management of this information system. The thesis is focused on Manufacturing Execution System (MES). The main objective of the thesis will draft measures and changes to optimize the information system. After introducing these changes should eliminate the weaknesses in the system and bring simplification of working with the system.

KLÍČOVÁ SLOVA

Informační systém, IS, MES, PHARIS, UNIS, optimalizace, informační výrobní systém, SQL, Webová aplikace, proces, testování

KEY WORDS

Information system, IS, MES, PHARIS, UNIS, optimization, Manufacturing Execution System, web application, proces, testing

BIBLIOGRAFICKÁ CITACE

HLAVÁČ, P. *Posouzení informačního systému firmy a návrh změn*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2017. 95 s. Vedoucí diplomové práce doc. Ing. Miloš Koch CSc.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 22. května 2017

.....

podpis

PODĚKOVÁNÍ

Rád bych poděkoval panu doc. Ing. Miloši Kochovi Csc., za čas, který mi věnoval, a za rady a připomínky, které mi dopomohly k napsání této práce. Rád bych také poděkoval panu Jaromíru Šánkovi a panu Leoši Honsovi za odbornou konzultaci mé diplomové práce.

OBSAH

ÚVOD.....	11
VYMEZENÍ PROBLÉMU A CÍLE PRÁCE	12
Vymezení problému.....	12
Cíle práce	12
2 TEORETICKÁ VÝCHODISKA.....	13
2.1 Informace a data.....	13
2.2 Informační systém.....	15
2.2.1 Význam IS	15
2.2.2 IS/ICT	15
2.2.3 Struktura IS	17
2.2.4 Klasifikace IS.....	18
2.2.5 Výrobní informační systém	20
2.2.6 Procesy v informačním systému	21
2.2.7 Hodnocení IS	21
2.3 Databáze.....	22
2.3.1 Databázový systém	22
2.3.2 Jazyk SQL.....	22
2.3.3 Datové modely.....	23
2.3.4 Entita a atributy.....	23
2.3.5 Relace.....	24
2.3.6 Klíče relace	24
2.3.7 Integritní omezení	24
2.3.8 Kardinalita vztahů.....	25
2.3.9 Normalizace	25

2.4	Webová aplikace	26
2.5	Bezpečnost	27
2.5.1	Bezpečnost webových aplikací	27
2.5.2	Bezpečnost IS	27
2.5.3	Úroveň bezpečnosti.....	29
2.6	Moderní trendy v informačních systémech.....	30
2.6.1	Outsourcing.....	30
2.6.2	Cloud.....	30
2.7	Reportovací služby.....	31
2.8	Vývoj IS	31
2.9	Testování	32
2.9.1	Automatizované testování.....	33
3	ANALÝZA SOUČASNÉ SITUACE	34
3.1	Představení společnosti	34
3.1.1	Předmět podnikání	34
3.1.2	Úspěchy společnosti	35
3.2	Organizační schéma společnosti	37
3.2.1	Organizační struktura.....	38
3.3	Hospodářské výsledky společnosti	39
3.4	Informační systém PHARIS.....	41
3.4.2	Zabezpečení ve společnosti	43
3.5	Modul Výrobní zdroje.....	43
3.6	Testování	48
3.6.1	Proces vytvoření pracovního kalendáře	49
3.7	Analýza systému	50
3.7.1	SWOT	50

3.7.2	HOS8	52
3.8	Zhodnocení analýzy	54
4	VLASTNÍ NÁVRHY A ŘEŠENÍ	55
4.1	Testování systému	55
4.1.1	Testovací scénáře	57
4.1.2	Manuální testování	60
4.1.3	Automatizované testování	61
4.1.4	Reportování nalezených chyb při testování	65
4.1.5	Procesy v systému	69
4.2	PHARIS Wiki	72
4.2.1	Pracovní kalendáře	73
4.2.2	Zařízení	73
4.2.3	Výrobní model	74
4.2.4	Události	74
4.2.5	Řídící programy	75
4.3	Mobilní aplikace	76
4.4	Ekonomické zhodnocení	79
4.4.1	Náklady a přínosy	79
ZÁVĚR		81
SEZNAM POUŽITÉ LITERATURY		82
SEZNAM OBRÁZKŮ		86
SEZNAM GRAFŮ		87
SEZNAM TABULEK		87
SEZNAM PŘÍLOH		87

ÚVOD

Kvalitní informační systém je klíčový prvek pro úspěch společnosti. Společnosti se snaží pořídit takový informační výrobní systém, aby byla práce s tímto systémem efektivní. Pořízení informačního systému také souvisí s nemalými investicemi a proto je velmi důležité, aby byl vybrán kvalitní informační systém. Tato diplomová práce se zabývá posouzením aktuálního stavu informačního výrobního systému a následným návrhem změn.

Diplomová práce je rozdělena na tři základní části. První část obsahuje teoretický základ a základní pojmy informačních systémů. V této části se čtenář seznámí se základním rozdělením informačních systémů, se základními pojmy databází, bezpečností a v neposlední řadě také se řízením projektů.

V následující části je představena společnost, která vyvíjí informační systém PHARIS a její významní zákazníci. Po představení společnosti bude provedena analýza informačního výrobního systému a definují aktuální problémy tohoto systému. Tato část také obsahuje analýzu podniku pomocí SWOT analýzy. Výstupem této části bude identifikace slabých míst systému a následné shrnutí zjištěných výsledků analýz.

Poslední část představuje vlastní návrh řešení. V této části jsou představeny řešení problémů, které byly identifikovány v předchozí části a je zde představen také návrh změn pro efektivnější práci s informačním výrobním systémem. V návrhové části je definován postup pro psaní testovacích scénářů, návrh procesů při testování a představení vytvořených skriptů pro automatizované testování. Další část popisuje doporučení pro webový portál PHARIS WIKI, který slouží jako uživatelská podpora pro zákazníky. V neposlední řadě je zde představen koncept pro rozšíření informačního výrobního systému PHARIS i na mobilní zařízení. Na závěr práce jsou představeny náklady a přínosy, které plynou z navržených řešení.

VYMEZENÍ PROBLÉMU A CÍLE PRÁCE

Vymezení problému

Informační výrobní systém PHARIS patří mezi nejprodávanější výrobní systémy v České republice. Pro udržení této pozice na trhu je potřeba vydávat stále nové verze systému, tak aby uspokojily požadavky zákazníků a byly atraktivní pro potenciální zákazníky.

Problémem informačního systému je regresní testování, díky velkému množství nových funkcionalit. Testování je nezbytné před uvedením každé nové verze systému do běžného provozu. Informační výrobní systém také nabízí uživatelskou nápovědu pomocí webového portálu PHARIS Wiki. Tento webový portál bohužel není kompletní a je potřeba doplnit informace u jednotlivých modulů systému. Informační systém také nepodporuje zobrazení na mobilních zařízeních.

Cíle práce

Cílem této diplomové práce je analýza a návrh změn informačního výrobního systému PHARIS, který je vyvíjen společností UNIS a.s.. Analýza informačního systému bude probíhat jak po stránce technické, tak po stránce uživatelské. Výsledek této analýzy bude sloužit jako podklad pro následný návrh opatření pro optimalizaci výrobního systému. Hlavním cílem práce bude dokončení skriptů pro automatizované testování modulu Výrobní zdroje, který je součástí informačního systému. Výstupem této diplomové práce bude návrh změn, který bude obsahovat řešení aktuálních problémů výrobního informačního systému PHARIS.

2 TEORETICKÁ VÝCHODISKA

2.1 Informace a data

V informačním systému jsou data velmi důležitou složkou. Tímto pojmem lze označit zvukový soubor, text, číslo, obrázky nebo video. Data nám vypovídají o reálném světě, popisují vlastnosti objektů a reprezentují fakta. Informace získáváme právě z dat, k těmto datům také potřebujeme další kontext nebo popis, jinak nám nemusí příliš dávat smysl.(1)

Data také obsahují několik specifických vlastností jako to, že se neustále mění, lze ověřit jejich přesnost a správnost, jsou obsáhlá a bývají nekompletní, proto se musí doplňovat nebo přidávat nová.(2)

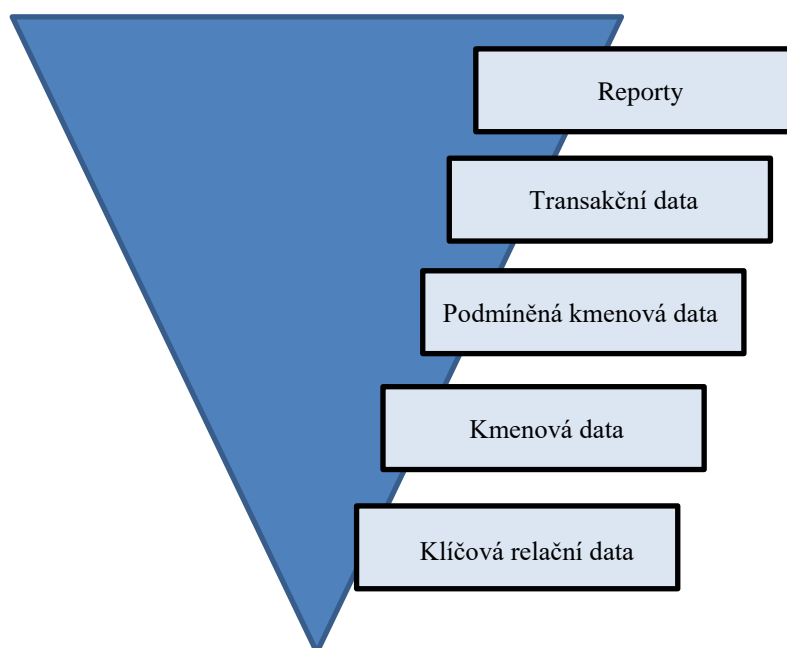
Data můžeme rozdělit do dvou kategorií :

- **Strukturovaná data** – data jsou zde uložena pomocí relačních databázových systémů. Díky tomuto systému uložení jsou data uložena tak, že je práce efektivnější a můžeme snadno vybírat konkrétní data.
- **Nestrukturovaná data** - jedná se o obrázky, videozáznamy nebo zvukové stopy. Sklenák je definuje jako „*Tok bitů bez dalšího rozlišení.*“(1, s. 2)

V podnikových informačních systémech reprezentují data údaje o významném faktu a lze je rozdělit jako:

- **Data o společenských podmínkách** – jsou to data o rozvoji nových služeb, produktů a technologií, úrovni ekonomického a demografického prostředí, produktivity nebo data o nákladech
- **Data o trhu** - data o konkurenceschopnosti, komoditním trhu nebo data o poptávce a nabídce
- **Interní data podniku** – data o fungování podniku, prognózy prodeje nebo podnikových zdrojích. Tato data slouží k definování okamžité reakce na změnu okolí. (3)

Kategorie podnikových dat:



Obrázek 1: Kategorie podnikových dat

Zdroj: (3)

První kategorií jsou klíčová relační data, tato data definují organizační strukturu a zachycují činnost podniku. Následují kmenová data, tyto údaje popisují vlastnosti zákazníků, dodavatelů, produktu nebo materiálu. Podmíněná klíčová data lze definovat jako specifické podmínky, které ovlivňují workflow nebo cashflow. Tyto podmínky se také podílí na cenotvorbě, dle charakteristik jednotlivých zákazníků. Příkladem kategorie transakčních dat jsou údaje o dopravě nebo počet kusů objednaného zboží. Reporty ve společnosti slouží k podpoře rozhodování a zobrazují souhrny určitého časového období, jako je například počet objednávek za jeden měsíc. (3)

Pojem informace lze chápat jako sdělení nějaké zprávy nebo poznatku. Takto informaci definuje Tvrdíková:

„Informace je název pro obsah toho, co si vyměňujeme s okolním světem, když se mu přizpůsobujeme a když na něj působíme svým přizpůsobováním.“ (4, s. 18).

Informace je často reprezentována jako zpráva, která interpretuje skutečnosti nebo vědomosti. Pokud chceme, aby byla informace efektivní, měla by být čitelná a srozumitelná, měla by také být ve správný čas na správném místě a neměla by obsahovat chyby.(3)



Obrázek 2: Řetězec informačního systému
Zdroj: (3)

2.2 Informační systém

2.2.1 Význam IS

Informační systém může podstatným způsobem ovlivnit celkovou hodnotu podniku, prodávané výrobky nebo konkurenceschopnost. Z tohoto důvodu je při zavádění velmi důležité položit si otázku, jaký bude mít zavedení IS význam pro podnik. (5)

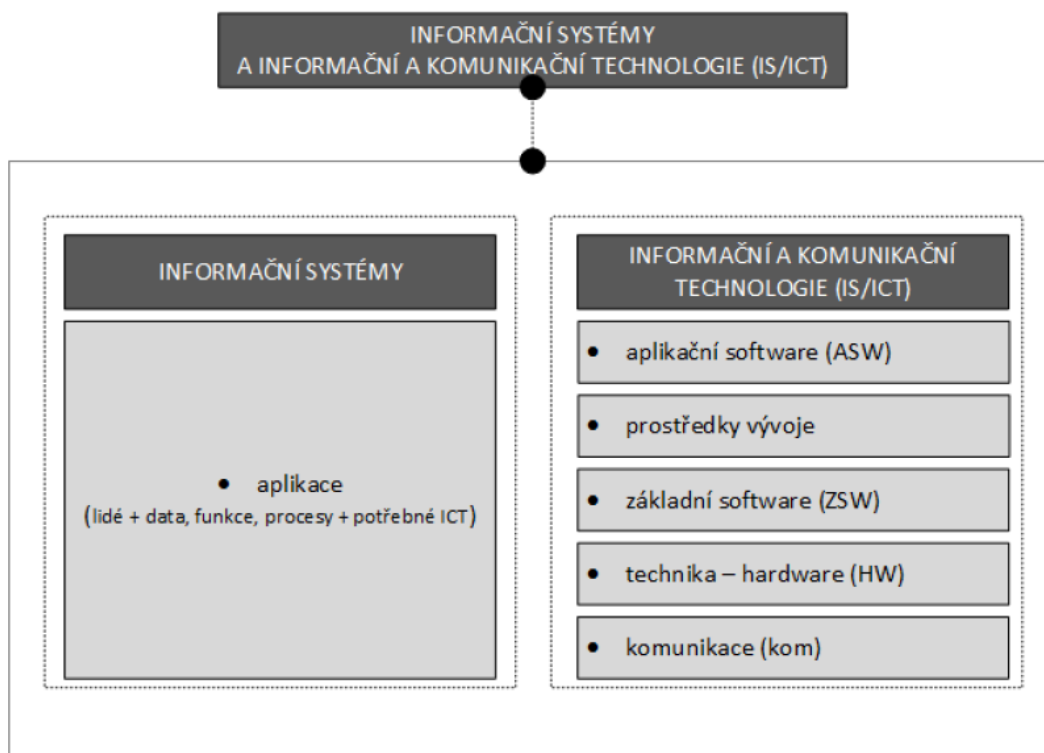
V případě že je informační systém nasazen a správně funguje, pak naplňuje podnikové cíle. Napomáhá ke zvyšování příjmů podniku a redukuje náklady. Dalším hlediskem je také následná kontrola, zda je nový systém efektivní a zda bylo dosaženo podnikových cílů.(5)

2.2.2 IS/ICT

Pojem IS/ICT (informační systémy a informační a komunikační technologie) nám definuje jakými nástroji a jakým způsobem je ve společnosti řízena informatika. Jedná se o množství datových, programových, aplikačních a technických bází a technik.(7)

Informační systém by měl splňovat 4 základní požadavky. IS by měl být:

- Pružný – IS se musí okamžitě přizpůsobovat novým požadavkům
- Udržovatelný – provoz musí být snadný na údržbu
- Bezpečný – bezpečnost je velmi důležitá, únik informací by mohl způsobit velké škody
- Spolehlivý – IS by měl být kvalitní, na plynulosti systému často závisí chod celé společnosti (8)



Obrázek 3: Základní vymezení IS/ICT
Zdroj: (7)

Definice informačního systému podle Sodomky:

„Podnikový informační systém vytvářejí lidé, kteří prostřednictvím dostupných technologických prostředků a stanovené metriky zpracovávají podniková data a vytvářejí z nich informační a znalostní bázi organizace sloužící k řízení podnikových procesů, manažerskému rozhodování a správně podnikové agendy.“ (6, s. 61).

2.2.3 Struktura IS

Informační systém obsahuje tyto části:

1) Hardware

Mezi hardware můžeme zařadit fyzické komponenty jako základová deska, procesor, grafická karta, zvuková karta, pevný disk a prvky počítačové sítě. Hardware spolupracuje se softwarem.(9)

2) Software

Software je definován jako sada instrukcí nebo programů. Tento pojem se používá k popisu počítačových programů. Mezi software můžeme zařadit nejrozumnější aplikace.

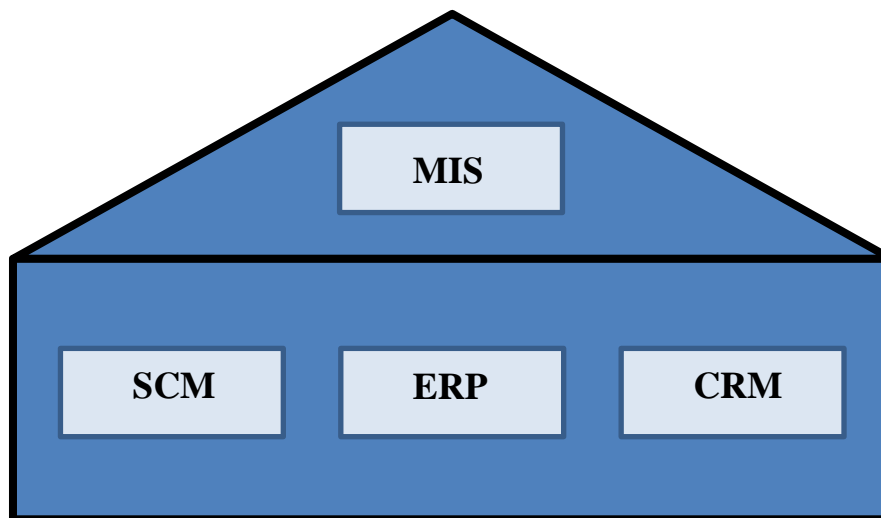
3) Orgware

Souhrn předpisů a metod nezbytných k dosažení definovaných cílů. Můžeme zde zařadit definice pro provozování a využívání informačních technologií. Orgware definuje uspořádání v informačním systému.

4) Peopleware

Jedná se o nejdůležitější část informačního systému, pokud je zanedbána, bude práce s informačním systémem neefektivní. Součástí této části jsou účinné prostředky pro adaptování člověka, jako efektivní část IS.(1)

2.2.4 Klasifikace IS



Obrázek 4: Rozdělení informačních systémů
Zdroj: (7)

MIS (Management Information system)

Manažerské informační systémy slouží k podpoře managementu při rozhodování. Tyto systémy získávají informace z ERP, CMR a SCM. Manažerské informační systémy obsahují datový sklad a poskytují zásadní informace, které musí být správně přiřazeny k určitým souvisejícím činnostem. MIS nám poskytuje služby pro podporu managementu: (6)

- Balanced Scorecard – podpora strategického řízení
- Document Manager – podpora práce s dokumenty
- Groupware – podpora spolupráce v týmu
- Enterprise Content Management – práce s nestrukturovanými informacemi

CRM (Customer Relationship Management)

Systémy zaměřené na řízení vztahu se zákazníky. Mezi procesy v tomto systému můžeme zařadit řízení kontaktů, marketing a servisní služby. V prostředí CRM používáme pojmy jako CRM koncepce a CRM procesy. Jejich cílem je uspokojení potřeby zákazníků a řízení ziskovosti. (6)

Koncepce CRM je rozdělena následně:

- 1 Globální CRM
- 2 Globální, lokálně uzpůsobená CRM
- 3 Lokální CRM

CRM procesy:

- Řízení obchodu – řízení obchodních činností
- Řízení marketingu - řízení marketingových kampaní
- Řízení kontaktů – komunikace mezi organizací a zákazníkem
- Servisní služby- pozáruční a záruční servis (6)

SCM (Supply Chain management)

Soubor procesů a nástrojů sloužících k dosažení maximální efektivity provozu a optimalizaci řízení v celém dodavatelském řetězci. SCM lze definovat jako propojení dodavatelů s odběrateli na základě informačních a komunikačních technologií. Díky tomuto propojení lze koordinovat celkový postup a sdílet informace.

Současná SCM informují zákazníka o stavu objednávky a snaží se snížit pravděpodobnost opoždění zásilky. SCM podporuje také plánovací činnosti jako napojení materiálových požadavků na možnost nákupu prostřednictvím elektronického tržiště nebo plánování požadavků v řetězci s ohledem na možnosti výroby a nákupu.(8)

SCOR definuje pro SCM tyto komponenty:

- Plán
- Nákup
- Výroba
- Expedice
- Reklamace

ERP (Enterprise Resource Planning)

Softwarové řešení pro hlavní podnikové procesy. Jako je expedice, výroba, prodej, účetnictví, nákup a výdej materiálu nebo řízení lidských zdrojů. ERP tyto procesy podporuje a automatizuje. ERP pokrývá většinu interních procesů, mezi které řadíme

výrobu, ekonomiku, personalistiku a logistiku. Model ERP je v současnosti nejpoužívanější model a využívá jej většina společností na světě.(8)

Systém vytváří data a automatizuje podnikové procesy. Provoz ERP by měl probíhat na architektuře klient/server. Zde je důležité zabezpečit komunikaci mezi klientem a serverem definováním přístupových práv a možnost hlášení nalezených chyb.(6)

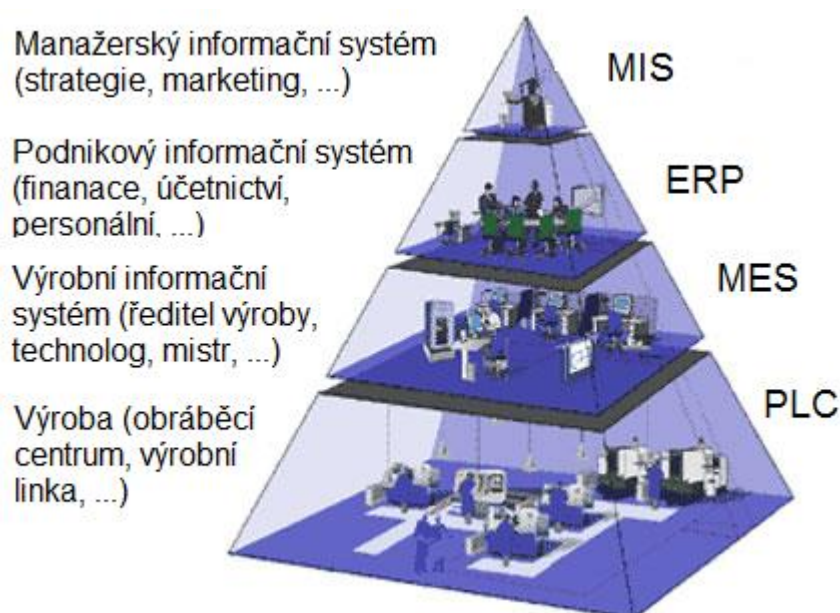
Klasifikace ERP systémů:

- All-in-one – systém pokrývající interní procesy
- Best-of-Breed – zaměření na specifické procesy
- Lite ERP – odlehčená verze ERP systému (6).

2.2.5 Výrobní informační systém

MES (Manufacturing Execution System) představuje návaznost informačního systému na systém výrobní. Výrobní informační systémy zajišťují sběr dat a jejich následné zpracování.(5)

Výrobní informační systémy tvoří vazbu mezi informačními systémy (ERP) a systémy sloužícími k automatizaci výroby.(10)



Obrázek 5: Kategorie informačních systémů
Zdroj: (10)

2.2.6 Procesy v informačním systému

Správná rozdělení procesů je v podnikových informačních systémech nezbytné. Procesy můžeme rozdělit dle činností. Mezi tyto činnosti můžeme zařadit výdej faktur, plánování zdrojů, zpracování nabídky nebo příjem materiálu na sklad. (8).

Podnikové procesy lze rozdělit na:

- Hlavní – jedná se o procesy, zabezpečující hlavní podnikové aktivity a podílí se na výkonnosti celého podniku.
- Řídící - pomocí těchto procesů definuje firma organizaci.
- Podporné - tyto procesy podporují procesy hlavní (11)

Pojem proces lze definovat jako soubor vzájemně souvisejících činností, které proměňují vstupy na výstupy. Tyto činnosti probíhají současně nebo na sebe navazují. Cílem procesů je transformovat vstupy tak, aby byly užitečné. Procesům lze také přiřadit parametry, které sledují jejich účinnost.(8)

2.2.7 Hodnocení IS

Informační systémy lze hodnotit dle rychlosti a kvality dosažených cílů. Pro hodnocení se využívají metody Balanced Scorecard (BSC) a Total cost of ownership (TCO) (5)

BSC (Balanced scorecard) – zde se zvolí systém hodnotících ukazatelů, který poskytuje zpětnou vazbu. Tato metoda popisuje význam měření výkonosti a umožňuje redefinovat cíle podniku. BSC zajišťuje rovnováhu mezi výstupy a ukazateli výkonosti.(6)

Sleduje podnik z těchto oblastí:

- Finanční perspektiva – umíme řídit potřebné zdroje?
- Zákaznická perspektiva – dokážeme pracovat se zákazníky?
- Procesní perspektiva – jsou naše procesy efektivně řízeny?
- Perspektiva učení a růstu – jsme schopni udržovat inovace v organizaci?(6)

TCO (Total cost of Ownership) – Tento ukazatel zahrnuje náklady na projekt po celou dobu jeho životnosti a to včetně skrytých nákladů jako jsou náklady na bezpečnost nebo náklady na dostupnost systému. Pomocí TCO lze porovnat výhodnost navrhovaných řešení.(12)

Celkové náklady lze rozdělit na:

- Celkové náklady na software
- Celkové náklady na pracovníky
- Celkové náklady na školení (12)

2.3 Databáze

2.3.1 Databázový systém

Databázi lze definovat jako sebedopisující kolekci integrovaných záznamů. K jejich popisu jsou použita metadata, tvořící systémový katalog. Jedná se o soubor nástrojů, díky kterým je zajištěno spolehlivé ukládání a následná manipulace s daty. Databáze bývají navrženy tak, aby bylo možno pracovat s velkými objemy dat. Pod pojmem databáze si lze představit data umístěná na databázovém serveru nebo lékařské záznamy.(14)

Databázový systém (DS) je tvořen databází a systémem řízení báze dat (SŘBD)

DS mohou být:

- Hierarchické a síťové
- Relační
- Objektové databázové systémy (13)

2.3.2 Jazyk SQL

Pomocí strukturovaného dotazovacího jazyku neboli SQL (Structured Query Language) můžeme přistupovat a pracovat s daty v databázi. Protože existuje spousta databázových systémů od různých výrobců, bylo potřeba standardizovat dotazovací jazyky. Výsledkem byl standart zvaný SQL. (14)

Příkazy byly rozděleny do následujících skupin:

- DML – příkazy pro manipulaci s daty
- DDL – příkazy pro definici struktury databáze
- DCL – příkazy pro řízení dat
- Ostatní příkazy – formulace pravidel

Mezi základní příkazy pro manipulaci s daty patří:

- INSERT – vložení dat
- SELECT – výběr dat
- UPDATE – změna dat
- DELETE – smazání dat (14)

2.3.3 Datové modely

Zde je důležitá podmínka, že pro každý typ datového objektu musíme navrhnout samostatnou datovou strukturu věty.

Typy datových modelů:

- Lineární datový model – není zde žádná vazby mezi tabulkami a lze je použít na libovolné databázi např. kartotéka pacientů
- Relační datový model – nejvyužívanější datový model, který je vytvořen několika lineárními datovými modely, propojenými pomocí klíče relace.
- Objektový datový model – pracuje se základním objektem, který odpovídá pojmu věta a obsahuje metody, které určují chování objektu (15)

2.3.4 Entita a atributy

Entita je základním prvkem ER diagramu, je to objekt reálného světa, který je reprezentován tabulkou a výčtem vlastností nazvané atributy (14)

Atribut tvoří hlavní zdroj dat v databázi a určuje vlastnosti každé entity. Rozlišujeme několik typů:

- Jednoduchý atribut – je složen z jednoho údaje
- Vícehodnotový atribut – složen z více údajů
- Odvozený atribut – lze jej vypočítat na základě jiného atributu, např. věk lze vyvodit z data narození.
- Speciální atribut – jedná se o cizí nebo primární klíč. Slouží k jednoznačné identifikaci. Za pomoci cizího klíče jsou vytvářeny relace mezi entitami (14)

2.3.5 Relace

Pojem relace lze definovat jako „*množina smysluplných spojení mezi entitami*“ (14, s. 157). Každé spojení musí být označeno a také musí být jednoznačně identifikovatelné. Relace bývá znázorněna jako šipka a slovně popsána. (14)

2.3.6 Klíče relace

Pro dodržení podmínky, že má být každý záznam v tabulce jedinečný, určíme atribut, který nám zaručí tuto jedinečnost (14)

Druhy klíčů relace:

- Kandidátní klíč – atribut, jehož hodnota určuje pouze daný záznam. Z kandidátních klíčů se vybírá jeden primární.
- Primární klíč – musí splňovat podmínky minimality a jednoznačnosti.
- Cizí klíč – je to atribut, který je obsažen v tabulce a odpovídá kandidátnímu klíči z cizí tabulky. (14)

2.3.7 Integritní omezení

Při modelování dat reálného světa dochází k určitým omezením teoretického modelu. Musíme tedy zajistit správnost a konzistenci dat v databázi.

Používáme následující úrovně integrity:

- Entitní integrita – relace má primární klíč, tento primární klíč identifikuje každý z řádků relace
- Referenční integrita – vytváříme relace pomocí cizího klíče společně s primárním klíčem v jiné tabulce
- Doménová integrita – atribut relace musí být z množiny přípustných hodnot (15)

2.3.8 Kardinalita vztahů

Kardinalita nám uvádí kolik n-tic relací sobě navzájem odpovídá.

- 1:1 – jedné n-tici relace odpovídá jedna n-tice jiné relace.
- 1:N – vždy jedné n-tici relace odpovídá více n-tic jiné relace.
- M:N – několika n-ticím relace odpovídá více n-tic relace jiné (15)

2.3.9 Normalizace

Normalizace je dekompozice relace do vhodnějšího tvaru, při zachování bezztrátovosti při zpětném spojení, odstranění redundance a zachování vzájemných závislostí.

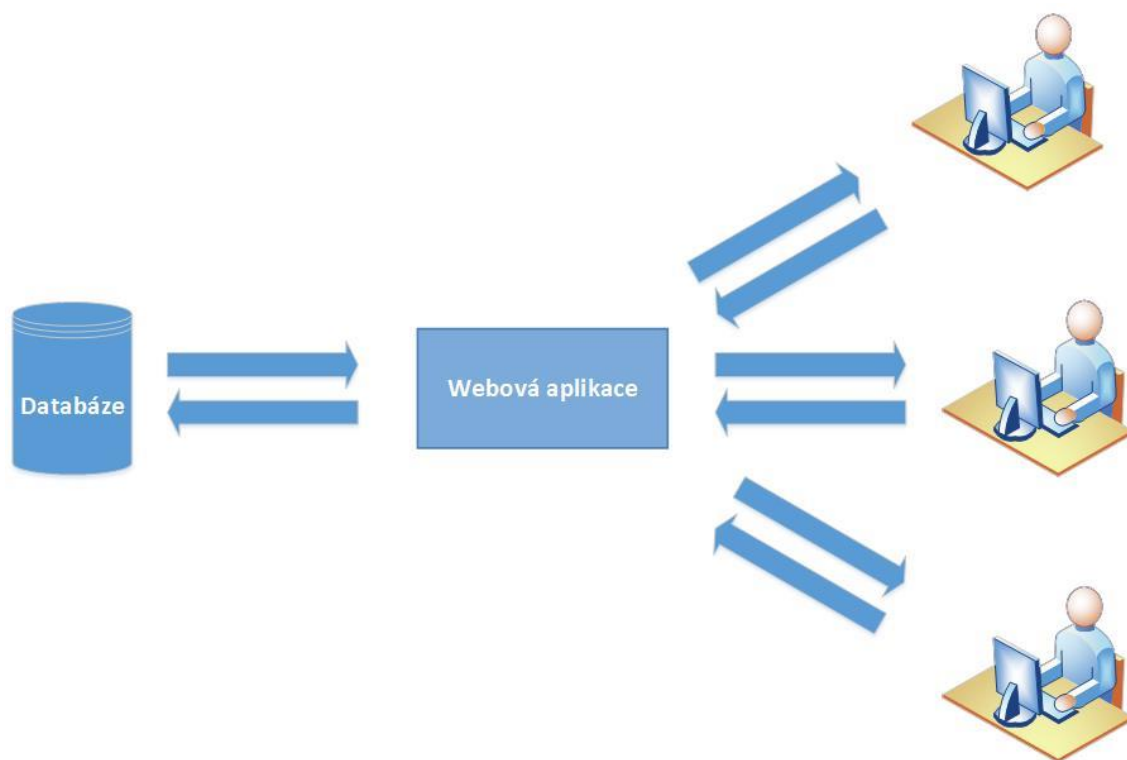
- **první normální forma** – všechny atributy relace jsou definovány nad skalárními obory hodnot
- **druhá normální forma** – relace je ve druhé normální formě, jestliže je v první normální formě. Atributy jsou závislé na kandidátním (primárním) klíči.
- **třetí normální forma** – relace musí být ve druhé normální formě a všechny neklíčové atributy jsou vzájemně nezávislé.
- **Boyce-Coddova normální forma** – varianta třetí normální formy, pokud je relace v této formě, je současně ve třetí normální formě. Relace je v Boyce-Coddově normální formě, pokud není žádná funkční závislost mezi kandidátními klíči, relace má více kandidátních klíčů, nejméně 2 z kandidátních klíčů jsou složené a kandidátní klíče se musí překrývat v některých attributech.
- **čtvrtá normální forma** – všechny vícehodnotové závislosti jsou zároveň funkčními závislostmi z kandidátních klíčů a relace je v Boyce-Coddově normální formě

- **pátá normální forma** – je velmi specifická a používá se ve výjimečných případech. Je případ spojené závislosti, která vyjadřuje cyklické omezení (15).

2.4 Webová aplikace

Pojem webová aplikace lze definovat jako komunikaci mezi žadatelem o službu a poskytovatelem služby. Při této komunikaci se využívá infrastruktura sítě Internet a přenosových protokolů. Jedním z těchto protokolů je Hypertext Transfer Protocol (HTTP), který je protokolem World Wide Web (WWW). Mezi další protokoly lze uvést SMTP (Simple Mail Transfer Protocol) pro přenos zpráv elektronické pošty nebo FTP (File Transfer Protocol) (16)

Jednou z vlastností je také distribuce dokumentů. Uživatelé tak mohou upravovat nebo číst dokumenty v reálném čase. Aktivitu jednotlivých uživatelů jsou také zaznamenávány. Webové aplikace umožňují přístup několika uživatelům ke zdroji informací najednou. Tyto aplikace také mohou rozřadit uživatele do různých bezpečnostních rolí s předem definovanými přístupovými právy. Webová aplikace je propojena s databází, do které jsou ukládána data, tato data jsou zobrazeny na vyžádání uživatele. Webová aplikace využívá webový prohlížeč.(17)



Obrázek 6: Webová aplikace

Zdroj: (17)

2.5 Bezpečnost

2.5.1 Bezpečnost webových aplikací

Ochrana klíčových informací společnosti je nezbytná a je nutné, aby byla věnována dostatečná pozornost tomuto problému.

Klíčové pilíře bezpečnosti jsou :

- autorizace
- autentizace
- integrita
- důvěrnost
- kryptografická ochrana
- ochrana před útoky
- session management
- konfigurace infrastruktury(18)

Webový aplikační firewall (WAF)

Jedná se o zařízení, které zajišťuje ochranu před útoky na aplikační úrovni. Webový aplikační firewall zvyšuje celkovou podnikovou bezpečnost, monitoruje a analyzuje provoz HTTP.(18)

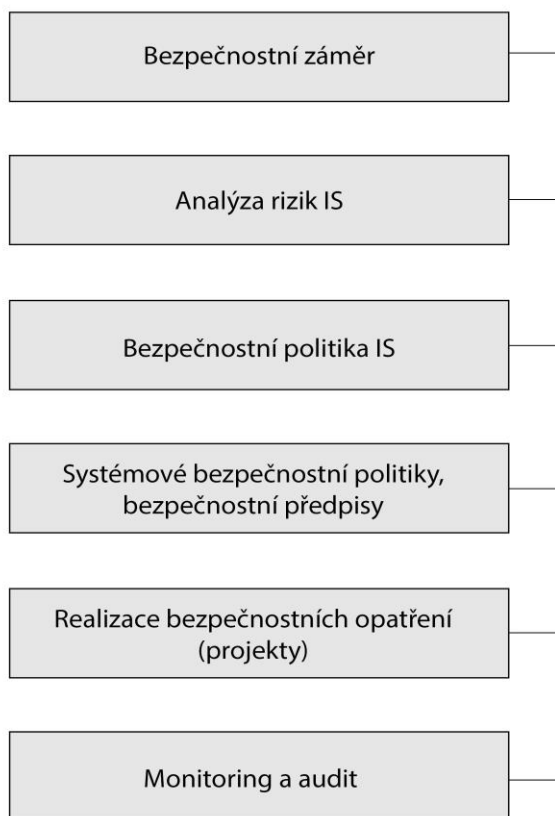
2.5.2 Bezpečnost IS

Bezpečnost informačních systémů je oblast, která je zaměřena na bezpečnost informací během celého životního cyklu. Velmi důležité je zde zachování:

- Dostupnosti – prevence ztráty přístupu k datům
- Důvěrnosti – prevence neautorizovaného prozrazení dat
- Integrity – prevence neautorizované změny dat

Ve společnostech se často zaměřují pouze na dostupnost. Úroveň integrity a důvěrnosti se řeší až po nečekaném vzniku incidentu (19)

Proces zabezpečení IS



Obrázek 7: Proces zabezpečení IS
Zdroj: (19)

Bezpečnostní záměr – pověřený útvar předloží bezpečnostní záměr, který obsahuje jak bude informační bezpečnost řešena a jakých cílů má být dosaženo

Analýza rizik IS – na základě bezpečnostního záměru se vypracuje analýza rizik, ta obsahuje zjištěná rizika, aktuální stav a návrhy opatření jednotlivých rizik

Bezpečnostní politika IS – po vyhodnocení analýzy rizik je vypracován dokument BPIS, dokument obsahuje další možné aktivity v oblasti bezpečnosti

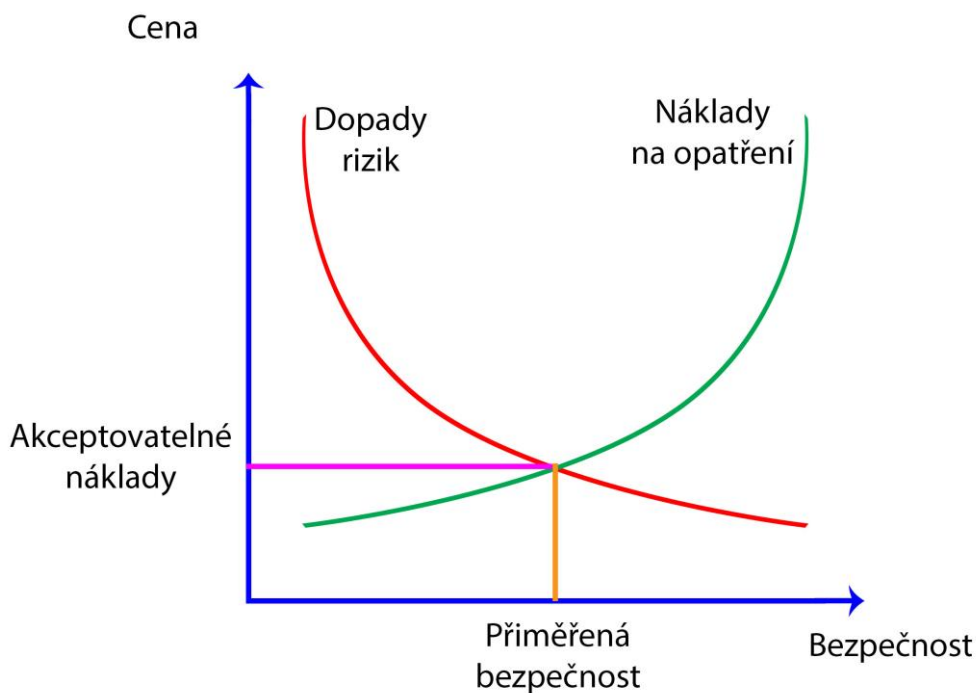
Systémové bezpečnostní politiky – bezpečnostní zásady z BPIS by se měli rozpracovat v dalších bezpečnostních směrnících

Realizace bezpečnostních opatření – postupná realizace

Monitoring a audit – po zavedení opatření musíme provádět pravidelné kontroly (19).

2.5.3 Úroveň bezpečnosti

Zaměření na hledání opatření, která jsou přiměřena hodnotě předmětu ochrany. Mezi tyto předměty ochrany řadíme informace. Společnosti by měli určit hodnocení informací a tomuto hodnocení přizpůsobit úroveň ochrany (18)



Obrázek 8: Zvážení rizika a opatření
Zdroj: (18)

2.5.4 Kybernetická bezpečnost

Souhrn organizačních, technických, právních a vzdělávacích prostředků k zajištění ochrany kybernetického prostoru.

- **Kybernetický prostor** – prostředí umožňující výměnu vznik a zpracování informací
- **Kybernetická bezpečnost** - souhrn organizačních, technických, právních a vzdělávacích prostředků k zajištění ochrany kybernetického prostoru.
- **Kybernetická bezpečnostní událost** – jedná se o událost, která může narušit bezpečnost informací
- **Kybernetický bezpečnostní incident** - jedná se o událost, která způsobila narušení bezpečnosti informací (20).

2.6 Moderní trendy v informačních systémech

2.6.1 Outsourcing

Zajištění procesů, služeb nebo činností, které jsou prováděny mimo organizaci. Jednotlivé segmenty jsou řešeny externím dodavatelem a to na základě Service Level Agreement (SLA) neboli dohody o úrovni poskytovaných služeb

K outsourcingu lze přistoupit z důvodu:

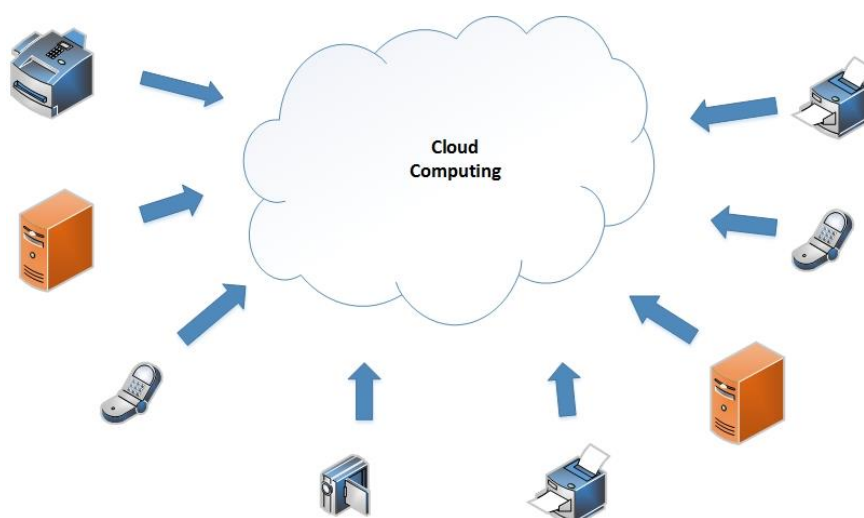
- Snížení nákladů
- Nedostatek vlastních lidských zdrojů
- Přenesení rizik na poskytovatele (21)

2.6.2 Cloud

Jedná se o službu používání hardwarových a softwarových nástrojů pomocí internetu. Tyto služby začínají v posledních letech využívat i největší poskytovatelé softwarových nástrojů.

Zákazník nemusí vlastnit žádné SW nebo HW nástroje. Tyto nástroje mu poskytuje poskytovatel pomocí služby. Zákazník se nemusí starat o infrastrukturu nebo provoz a platí pouze za využití zdroje.

Cloud se rozvíjí díky stabilnímu a rychlému internetu a také možnosti virtualizace na straně poskytovatelů IT. (22)



Obrázek 9: Cloud computing

Zdroj: (Vlastní)

Cloud zahrnuje tyto úrovně:

- PaaS (Platform as a Service) – služba poskytující kompletní prostředky pro vývoj a údržbu aplikací.
- IaaS (Infrastructure as a Service) – pronájem výpočetního výkonu, datových úložišť nebo hardwarové infrastruktury.
- SaaS (Software as a Service) – poskytování softwarových licencí. Software je zákazníkovi poskytován přes internet. (21)

2.7 Reportovací služby

Uživatelé systému by měli mít přístup k datům z databáze. Výstupy v podobě textového výpisu nemusí být dostačující, proto se využívají nástroje pro tvorbu reportů.

Tyto nástroje slouží k získávání znalostí dat a patří mezi bussines intelligence. Reporty se tvoří pomocí nástroje SQL Server Reporting Services, tento nástroj je doplňkem MS SQL Serveru. Nejčastěji se využívají nástroje jako Report Builder nebo Report Designer. Report Designer je vývojářský nástroj zatímco Report Builder je určen pro běžné uživatele. (23)

Panel Report Data, který je součástí Report Designeru obsahuje tyto položky:

- Datové zdroje – připojení k jednotlivým databázím
- Parametry – slouží k filtraci dat
- Datové sady – seznam datových sad(23)

2.8 Vývoj IS

Vývoj informačních systémů se řídí podle předem stanovených cílů. Pro vytvoření informačního systému je nezbytné mít informace o daném oboru, pro který je informační systém vyvíjen. Informační systém by měl být systematicky vyvíjen s ohledem na rozvoj podniku.

IS lze vyvíjet interně (IT oddělení) nebo externě (specializovaná společnost), nezbytná je ovšem komunikace mezi implementátorem a zadavatel po celou dobu realizace.

Každý projekt prochází následujícími fázemi:

- **Zahájení projektu**

Na první osobní schůzce se setkají zadavatel a implementační společnost. Na této schůzce se ujasní specifické požadavky zadavatele a implementační společnost následně vypracuje studii proveditelnosti.

- **Analýza požadavků**

V této fázi se vytváří detailní specifikace systému. Vytvoří se definice požadavků a funkční popis. Tyto dokumenty jsou důležitou součástí celého projektu. Nedůkladné provedení této části může vést k přepracování již implementované části a prodražit celý projekt.

- **Návrh řešení a implementace**

Implementační strana navrhne systém dle definovaných požadavků. V další části by měl být systém nasazen na testovací server.

- **Testování a předání projektu**

V této fázi se testuje nasazený informační systém. Skupina testerů zjišťuje nedostatky nebo chyby systému. Při úspěšném testování je informační systém připraven k předání zadavateli (24).

2.9 Testování

Testování softwaru se charakterizuje jedním ze tří pojmů: testování černé, bílé a šedé skříňky. Při testování černé skříňky tester netuší, jak software pracuje uvnitř. Zadá pouze údaje na vstupu a získá výstupní údaje. Netuší přitom proč se takovýto výsledek objevil. Při testování bílé skříňky má tester přístup ke zdrojovému kódu programu. Díky tomuto přístupu dokáže tester lépe přizpůsobit další testování. Kombinace černé a bílé skříňky je nazývána jako testy šedé skříňky. (31)

2.9.1 Automatizované testování

V případě testování musíme v některých případech provést několik tisíc opakovacích příkladů. Provést tyto opakování pomocí manuálního testování může být obtížné. Řešením jsou automatizované nástroje, které poskytují efektivnější způsob testování, než je manuální testování.

Mezi nejdůležitější vlastnosti testovacích nástrojů a automatizace testů můžeme zařadit:

- **Rychlost**

Nástroje automatizovaného testování jsou schopny provádět testovací případy 12krát, 200krát nebo i 1000krát rychleji než manuální testování. V určitých případech zabere testový případ v průměru pouze několik sekund. Manuální testování by v případě provedení několika stovek testovacích scénářů, trvalo podstatně delší dobu než automatizované testování.

- **Efektivita**

Automatizované testování také ve velké míře zkrátí čas potřebný k provedení testů. Tester tedy získá více času na promyšlení testovacích případů a plánování testů.

- **Správnost a přesnost**

Při manuálním testování začne tester po vyzkoušení několika stovek testovacích případů ztrácet pozornost. Z tohoto důvodu je výhodnější využívat automatizované testy, při kterých testovací nástroj vykoná stejnou práci a zkontroluje výsledky dokonale.

- **Neúnavnost**

Při automatizovaném testování můžeme konstatovat, že testy mohou být spouštěny stále dokola. Pokud bychom opakovali stále dokola testy pomocí manuálního testování, tester by se velmi rychle unavil. (31)

3 ANALÝZA SOUČASNÉ SITUACE

3.1 Představení společnosti

3.1.1 Předmět podnikání

Obchodní firma: UNIS, a.s.
Adresa: Jundrovská 1035/33
Komín
624 00 Brno
Česká republika

Identifikační číslo: 005 32 304
Právní forma: Akciová společnost

Datum zápisu: 14. května 1990

Společnost zapsaná v obchodním rejstříku vedeném Krajským soudem v Brně v oddílu B, vložka 5611.

Právní forma společnosti je akciová společnost. 204 ks akcie na jméno v listinné podobě ve jmenovité hodnotě 500 000,- Kč. Výše základního kapitálu je 102.000.000,- Kč (26).



Obrázek 10: Logo společnosti
Zdroj: (27)

3.1.2 Úspěchy společnosti

KOH-I-NOOR PONAS s.r.o. Polička - součást KOH-I-NOOR MACHINERY a.s.



Úkolem výrobního informačního systému MES PHARIS je řízení výroby lisovny plastů. Systém také zajišťuje sběr dat ze strojů a dalších zařízení, vizualizací, správy nástrojů nebo řízení údržby. V projektu je také zahrnuto řešení oboustranné komunikace s ERP K2. Výroba této společnosti je zaměřena na technické výlisky z oblastí zdravotní techniky, kosmetiky a elektrotechnického průmyslu.(28)

ROBERT BOSCH, spol. s r.o. České Budějovice



Díky nasazení MES PHARIS byl zabezpečen komplexní sběr procesních parametrů ze vstřikovacích lisů a vizualizaci výroby. Společnost se podílí na vývoji komponentů do osobních aut jako jsou plynové pedály, sací moduly, rozvaděče paliva a nádržové čerpadlové moduly. Mezi odběratele patří významné japonské a evropské automobilky.(28)

Kasko spol. s r.o. Slavkov – referenční instalace



Informační výrobní systém zabezpečuje řízení výroby lisovny plastů, řízení údržby, vizualizace, řízení kvality a další. Jedná se o referenční pracoviště systému PHARIS. Součástí řešení projektu je také propojení informačního výrobního systému s ERP HELIOS. Jedná se o společnost zabývající se vstřikováním plastových výlisků a výrobou forem.(28)

Continental Automotive Czech Republic s.r.o., závod Frenštát pod Radhoštěm



V tomto projektu MES PHARIS řeší zabezpečení komunikace se vstřikovacími lisy, vizualizace, správu nástrojů a zajištění sběru procesních parametrů. Výroba je v tomto závodě zaměřena na výrobu komponentů pro automotive.(28)

LUKOV Plast, spol. s r. o., Český Dub



MES PHARIS zde primárně zabezpečuje řízení výroby montážních linek slunečních clon a vstřikovny plastů. Projekt byl následně rozšířen o oblast Kapacitního plánování. Součástí řešení je také komunikace s ERP SAP. Společnost je zaměřena na výrobu technických a plastových elektro dílů.(28)

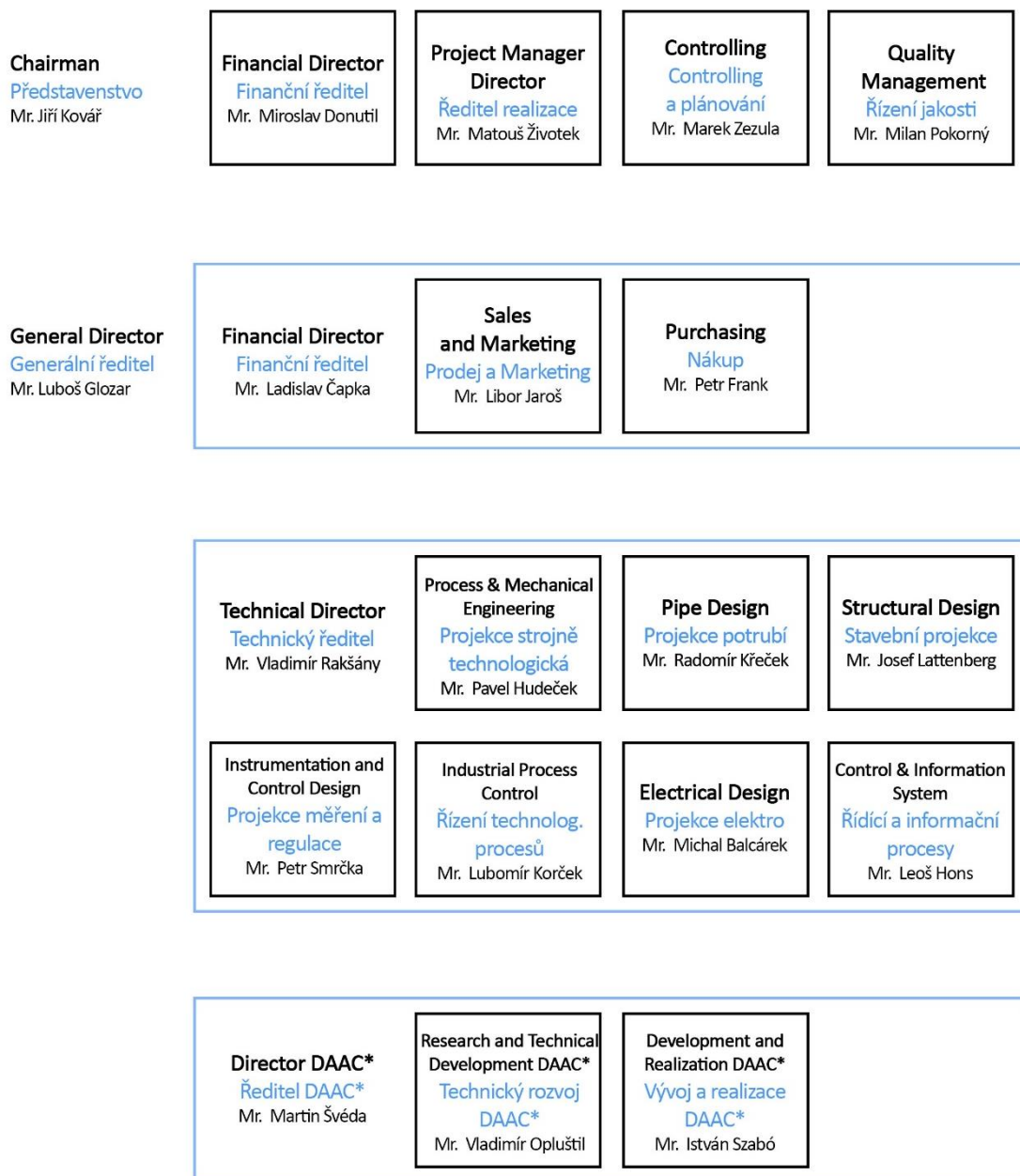
ATEK, s. r. o., Moravská Třebová



Informační výrobní systém zde zabezpečuje oboustranou komunikaci s ERP OR-SYSTEM. Systém pokrývá řízení výroby lisovny plastů a montáží a komplexně řeší jednoznačné identifikace a to včetně traceability. Společnost se zabývá vývojem a výrobou vstřikovacích nástrojů. (28)

3.2 Organizační schéma společnosti

ORGANIZAČNÍ SCHÉMA SPOLEČNOSTI

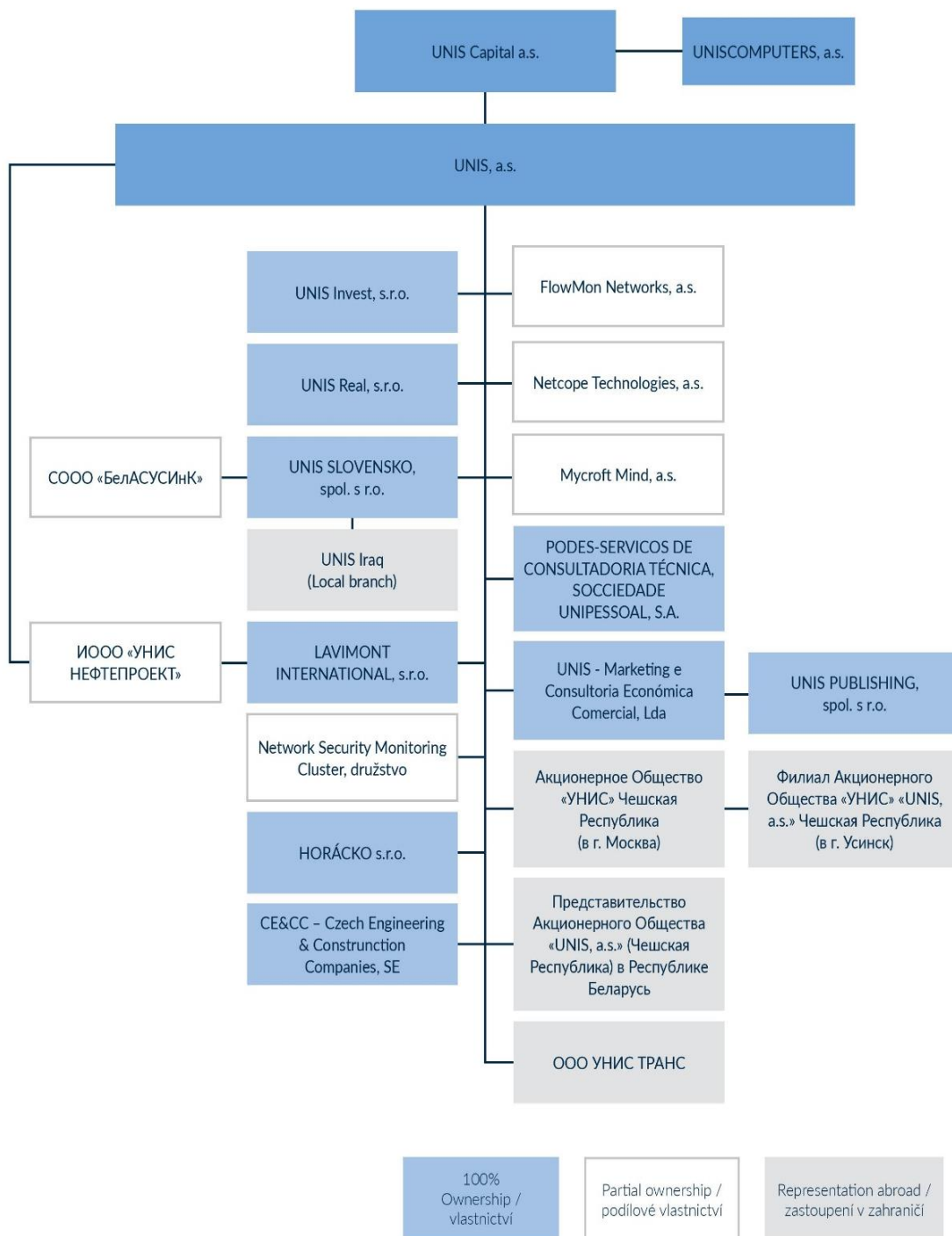


Obrázek 11: Organizační schéma společnosti
Zdroj: (30)

*DAAC – Division of Aerospace and Advanced Control

3.2.1 Organizační struktura

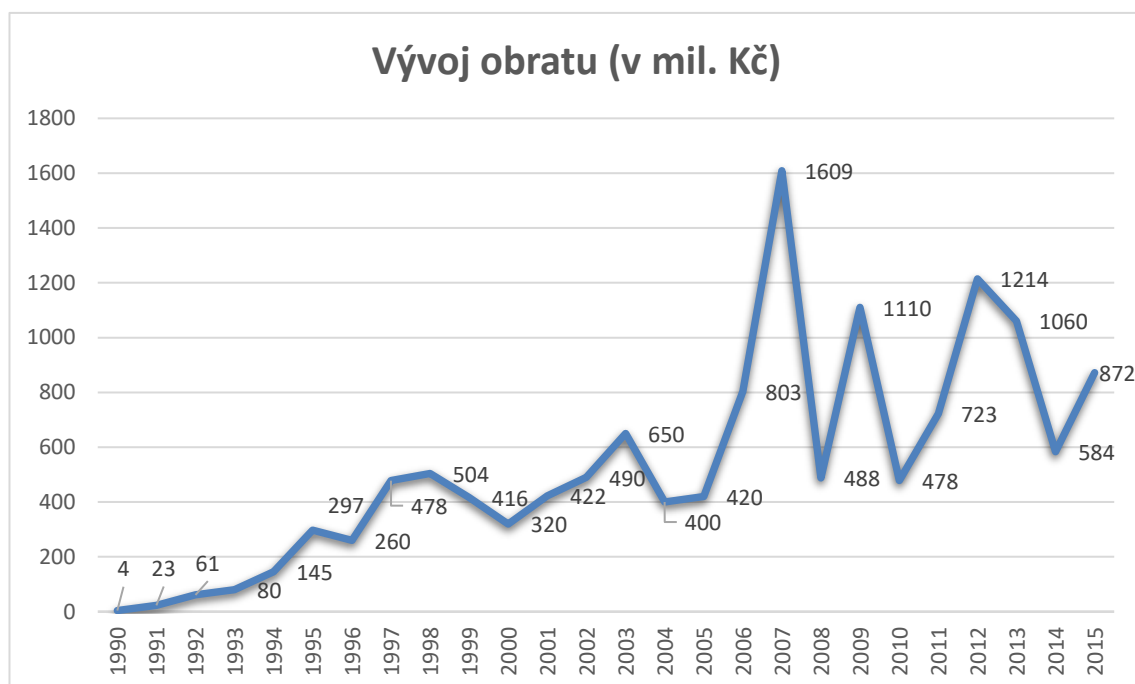
COMPANY STRUCTURE STRUKTURA SPOLEČNOSTI



Obrázek 12: Struktura společnosti

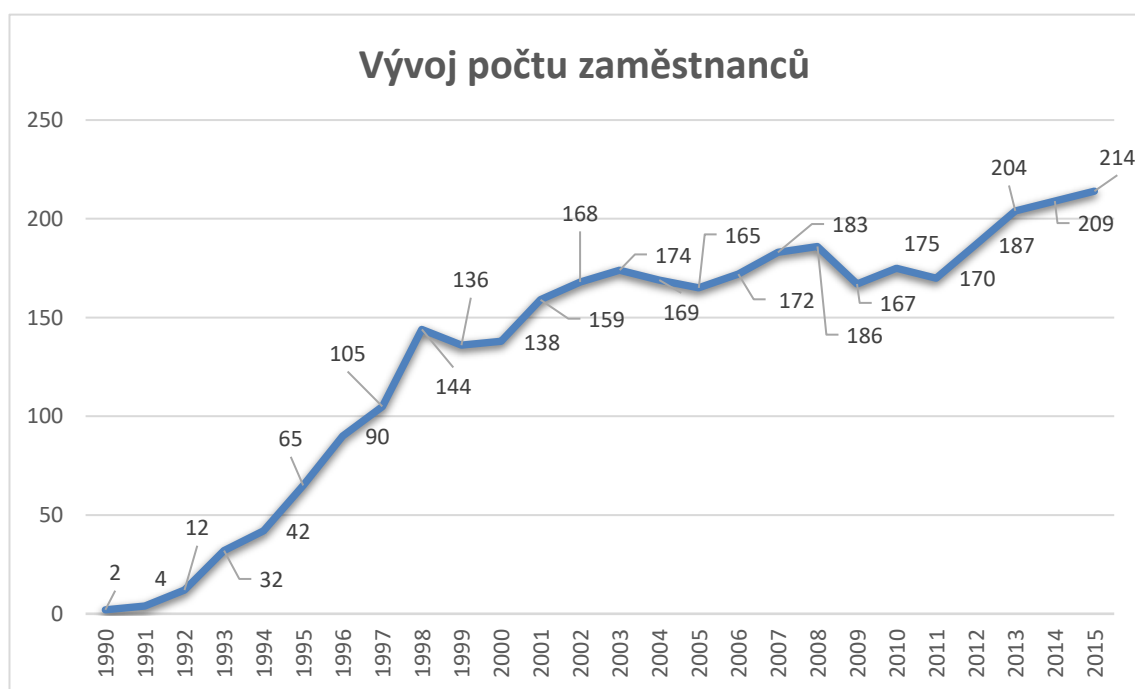
Zdroj: (30)

3.3 Hospodářské výsledky společnosti



Graf 1: Vývoj obrátu (v mil. Kč)

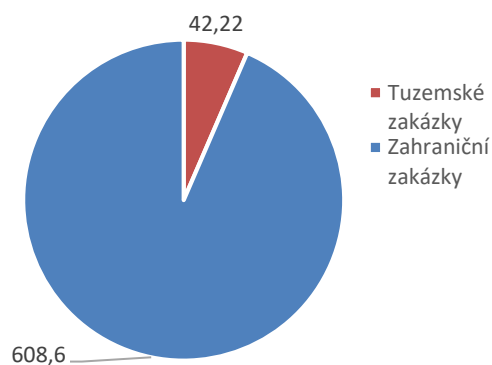
Zdroj: (30)



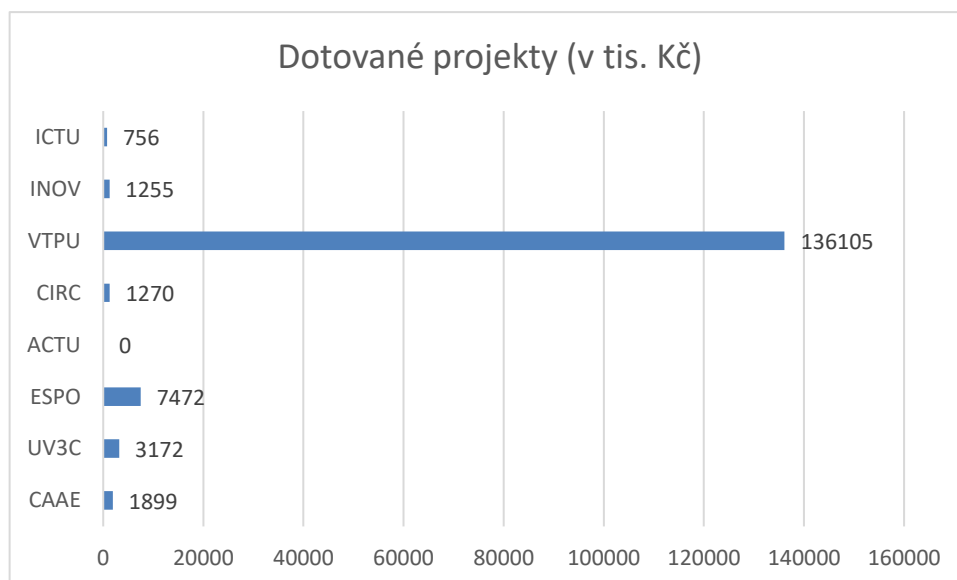
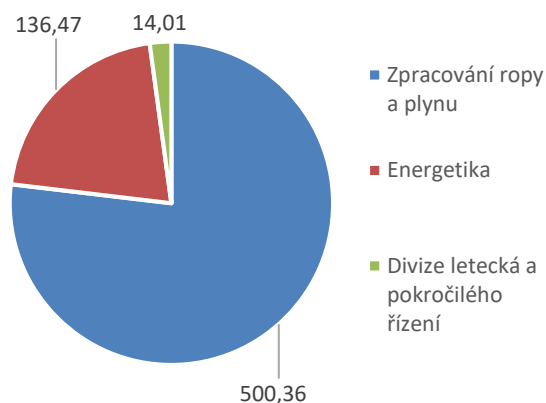
Graf 2: Vývoj počtu zaměstnanců

Zdroj: (30)

Podíl zakázek z pohledu
konečného uživatele
(tržby za vlastní výkony a
zboží v mil. Kč)



Podíl zakázek dle oborového
členění
(tržby za vlastní výkony a zboží
v mil. Kč)



Graf 3: Podíl zakázek a dotované projekty
Zdroj: (30)

ICTU - ICT and strategic services in company UNIS

INOV - Innovation project of company UNIS

VTPU - Scientific and Technical Park UNIS (VTP UNIS)

CIRC – Mobile Dedicated Equipment for Fulfilment of the Ability of Reaction of Computer Incidents

ACTU - Modular Electro Mechanical Actuators for ACARE 2020 aircraft and helicopters

ESPO – Efficient Systems and Propulsion for Small Aircraft

UV3C – Visual Computing Competence Centre

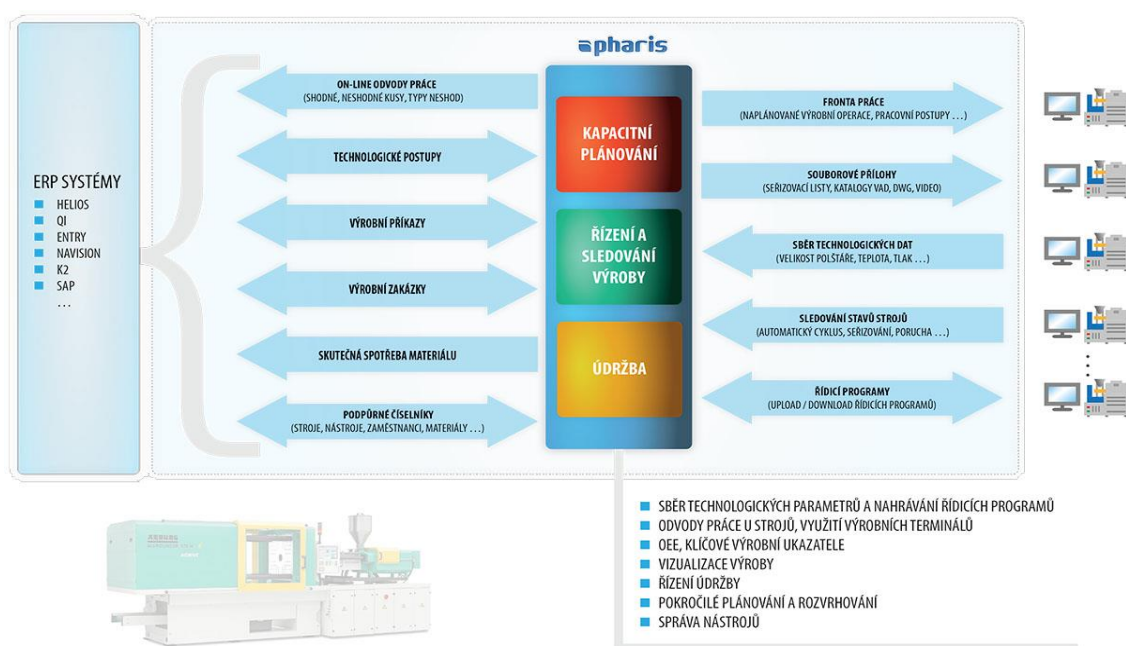
CAAE – Complex Affordable Aircraft Engine Electronic Control

3.4 Informační systém PHARIS

Výrobní informační systém PHARIS je plnohodnotný informační systém, který zajišťuje sledování a řízení výroby v reálném čase, sběr dat z technologických zařízení a ukládá veškeré průvodní informace. PHARIS poskytuje také nástroje pro detailní analýzu výroby. (29)

PHARIS disponuje uživatelsky velmi přívětivým prostředím. Základní menu systému je logicky organizováno do jednotlivých modulů. Mezi základní moduly informačního výrobního systému PHARIS patří Příprava výroby, Plán a rozvrh, Výroba, Výrobní logistika, Výrobní zdroje, Údržba, Monitoring, Balance a Ukazatele a Administrace.

Informační systém slouží ke sběru technologických parametrů a nahrávání řídicích programů, vizualizaci výroby, správě nástrojů, řízení údržby nebo pokročilé rozvrhování a plánování. Systém PHARIS také monitoruje využití výrobních terminálů nebo odvody práce u jednotlivých strojů.



Obrázek 13: ERP a Pharis

Zdroj: (29)

3.4.1 Typy klientů MES PHARIS

MES PHARIS využívá následující typy klientů:

- Hlavní klient (kancelářská webová aplikace, určen pro manažery)
- Výrobní klient (dotyková obrazovka, určeno pro operátory)
- Vizualizační klient (použití na velkoplošných obrazovkách)
- Výrobní mini klient (integrovaná čtečka RFID) (29)

PHARIS využívá SQL server a komunikuje s ostatními ERP systémy, se systémy řízení kvality a se systémy CAD/CAM. PHARIS je možné zobrazit na hlavním klientu jako webovou aplikaci, která je určena převážně pro kanceláře a pro manažery. Systém je pomocí internetové sítě propojen s výrobním klientem, určeným pro pracovníky ve výrobě, vizualizačním klientem, který je použit jako informační panel na obrazovkách, nebo mini klientem. V síti jsou také připojeny jednotlivé stroje jako jsou CNC stroje nebo vstřikovací lisy.



Obrázek 14: Klienti systému Pharis
Zdroj: (29)

3.4.2 Zabezpečení ve společnosti

Zabezpečení je řešeno na několika úrovních. Pokud se budeme bavit v rámci fyzické bezpečnosti, je společnost zabezpečena kamerovým systémem. Navíc je zde také povinnost zaměstnance identifikovat se pomocí čipové karty, pokud se tak nestane, je mu přístup odepřen. Každý zaměstnanec má definovaná práva pro přístup do určitých sekcí. V případě IT bezpečnosti se musí každý zaměstnanec vždy přihlásit do systému pomocí svého uživatelského jména a hesla. Jednotlivým uživatelům jsou také přiděleny přístupy. Uživatelské jméno a heslo zaměstnanec získá po osobní návštěvě IT oddělení.

3.5 Modul Výrobní zdroje

Informační systém PHARIS je velmi rozsáhlý systém a disponuje několika specifickými oblastmi. Diplomová práce je zaměřena na oblast Výrobní zdroje. Oblast Výrobní zdroje obsahuje pět modulů, kterými jsou Pracovní kalendáře, Zařízení, Výrobní model, Události a Řídící programy. V těchto pěti modulech jsou pokryty činnosti související s výrobními zdroji.

Pracovní kalendáře

Modul Pracovní kalendáře slouží k plánování a editaci pracovních kalendářů. V tomto modulu je možné zobrazit aktuální seznam pracovních kalendářů nebo historii pracovních kalendářů. Pracovní kalendáře jsou ukládány do databáze pracovních kalendářů. S databází pracovních kalendářů pracuje více oblastí systému PHARIS. Při plánování pracovních kalendářů lze nastavit denní pracovní dobu, počet pracovníků na jeden stroj a vytíženost stroje. Lze také přidat výjimečný kalendář. Vytvořený pracovní kalendář lze následně přiřadit k jednotlivému stroji nebo uživatelským skupinám.

Modul pracuje s databází pracovních kalendářů a jejich editace je velmi jednoduchá. Uživatel může vytvořit libovolný počet základních kalendářů u kterých je možné definovat disponibilitu. Systém umožňuje také vytvořit kalendář výjimečný. U těchto kalendářů bude v případě souběhu základního a výjimečného brána disponibilita z kalendáře výjimečného. Lze také definovat výjimky, výjimka definuje dostupnost jednoho dne a lze u ní nastavit opakovatelnost. Výjimka se přiřazuje ke kalendáři nebo ke zdroji. Kalendář může mít několik výjimek. Základní kalendář má neomezenou platnost.

Správa zařízení

V tomto modulu lze zobrazit zařízení ve společnosti, přiřadit zařízení do uživatelsky definovaných skupin nebo přidat zařízení dle typu zařízení. Při vytváření nového zařízení lze nastavit ID, název, popis, typ, umístění, RFID, údržbu, jednotky, počet výrobků a mnohem více dalších parametrů jednotlivého zařízení. Modul pracuje s databází zařízení a pro efektivní práci je nezbytné, aby společnost evidovala veškeré své stroje. Tyto zařízení lze také sledovat a zahrnout do plánování.

Správa zařízení

Zařízení Typy zařízení Skupiny zařízení

Detail zařízení

Identifikační číslo	Název zařízení	Popis zařízení	Výměna varianty
23154	Testovací	Testovací forma	Žádná

Typová příbuznost	Typ zařízení	Výrobní číslo (Interní kód)	Evidenční číslo
Forma	Forma	HP8897564	E58946458

Fyzické umístění	Datum pořízení	SOP (System operating procedure)	Sledovat	Zahrnout do plánování
E6	11.4.2017		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Registrační číslo	Číslo čárového kódu	RFID číslo
R87958468	8975468	R456585

Životnost [počet cyklů]	Životnost [h]	Název ve výrobním modelu	Umístění ve výrobním modelu	Údržba [počet cyklů]
8500	0,00			200

Počet připravených registrací k operaci: 0 Počet připravených registrací ke všem operacím: 0

Jednotky	Jednotky kalibrace	Předepsaná skladová pozice	Skladová položka
Skupina jednotek: Jednotky délky: mm	Skupina jednotek: Jednotky délky: mm		<input type="checkbox"/>

Počet výrobků: 0

Zařízení

Počet záznamů: 2082 10 záznamů na stránku

Název zařízení	Evidenční číslo	Typ zařízení	Typová příbuznost	Umístění	Popis zařízení	Zahrnout do plánování
00305		Stroj	Stroj		Hlubička Innovation	Ne
00JS Forma		Forma	Forma			Ne

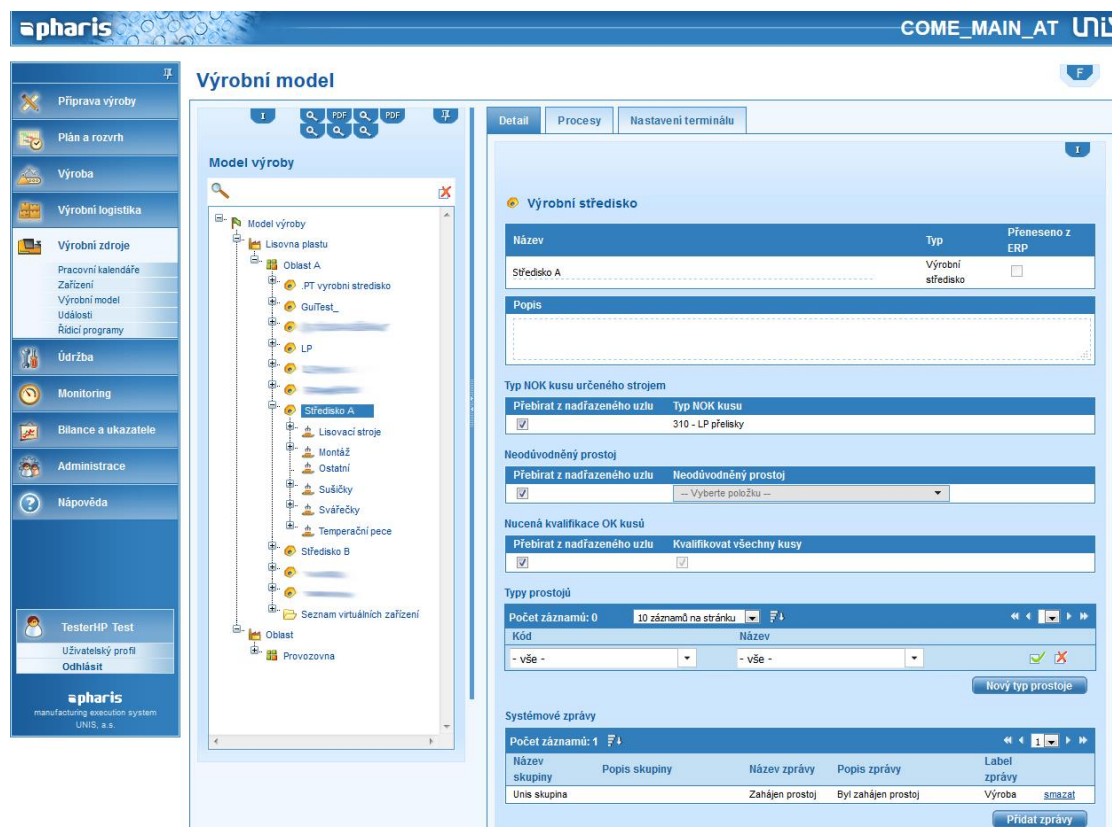
Obrázek 16: Pharos Správa zařízení
Zdroj: (Vlastní)

V modulu správy zařízení jsou zahrnuty také typy zařízení a skupiny zařízení. V oblasti typů zařízení lze definovat nové typy nebo přiřadit zařízením typy jako je forma, stroj, lis nebo mnoho dalších typů zařízení. V oblasti skupin zařízení lze jednotlivá zařízení přidělit do skupin, skupinám zařízení je poté možno přiřadit např. výrobní parametry nebo nastavení pro reporty.

Výrobní model

Sofistikovaný model výroby, ve kterém lze definovat nové podniky, provozovny, oblasti, výrobní střediska, výrobní jednotky nebo virtuální zařízení. Manažer systému má tak jedinečný pohled na stromovou strukturu celého systému, a je zde dohledatelné umístění veškerých strojů v jednotlivých oblastech výroby.

V tomto modulu je také velmi jednoduché zobrazit definované procesy. Mezi další užitečnou věc lze zařadit nastavení jednotlivých terminálů, konkrétněji zobrazení prvků na každém výrobním klientu, nastavení výchozí stránky nebo omezení výběru zařízení, nebo posílání zprávy po stažení výrobního programu. Ke každé oblasti, zařízení lze také přidat zodpovědného uživatele.



Obrázek 17: Pharis Výrobní model

Zdroj: (Vlastní)

Vytváření nové oblasti nebo zařízení probíhá vyvoláním nabídky pomocí pravého tlačítka myši. Po vyvolání této nabídky se zvolí Nové výrobní zařízení. Po zadání příslušného pojmenování zařízení je potřeba tuto operaci potvrdit tlačítkem enter.

Následně dojde k vytvoření požadovaného zařízení. Zařízení lze také vyhledávat pomocí zadání názvu zařízení a potvrzení tlačítkem enter.

Události

V modulu Události se nachází možnosti jako Plánované a probíhající události, Historie událostí, Reporty a Typy událostí. V plánovaných událostech se zobrazují jednotlivé události, tyto události se dají editovat nebo se mohou vytvořit nové události. Jednotlivé události jsou barevně odlišeny dle typu události. Dále jde zobrazit Historii událostí. V tomto modulu je také možné definovat nový typ události. Záložka Reporty obsahuje přehled reportů z oblasti výměny nástrojů. Reporty lze přizpůsobit dle požadavků zákazníka, nejen vzhled, ale i obsah.

Reporty si může vytvořit uživatel v externí aplikaci mimo MES PHARIS, která se skládá z následujících nástrojů:

- Report Manager – správa reportů,
- Report Builder – vytváření a editace reportů,
- Report Viewer – zobrazení reportů.

Typ události	Zařízení	Evidenční číslo	Stav	Plánované zahájení	Plánované ukončení	Zahájeno	Ukončeno
Údržba	KM350/19		Editace	13.06.2019 7:44:00			
Oprava	ZARZENI_STESTLD_NEMAZAT		Editace	10.04.2017 1:14:12			
Oprava	ZARZENI_STESTLD_NEMAZAT		Probíhá	08.03.2017 3:06:33		08.03.2017 3:06:33	
Porucha	STESTL32017022523100		Probíhá	25.02.2017 23:18:04		25.02.2017 23:19:30	
Údržba	STESTL32017022023050		Čeká na zahájení	20.02.2017 23:16:08	20.02.2017 23:17:08		

Obrázek 18: Pharis Události
Zdroj: (Vlastní)

Řídící programy

V této oblasti lze editovat a přidávat řídicí programy. K řídicímu programu lze přiřadit zařízení nebo připojit souborové přílohy.

Správa řídicích programů

Řídící programy

Počet záznamů: 20 10 záznamů na stránku

Název program	Popis	Verze	Platnost	Zařízení	Datum připojení souboru	Datum nahrání souboru	Připojena poznámka
DRIFT		1	Schváleno	NISSAN SKYLINE	04.12.2014 14:07:31		Ne
F0000 lis č. 16	zkouška lis č. 16	2	Schváleno	09-02-037-KM 300			Ne
F2516_25	F2516, 03106100, flansch Suzuk...	1	Schváleno	03-02-025-KM 160	12.12.2012 12:06:48		Ne
F2571 č. lisu 27	colimator stop	1	Schváleno	09-02-038-KM 200	04.10.2012 13:30:09		Ano
F2571_27	F2571, 078081/8200,	6	Neschváleno	09-02-038-KM 200			Ne
Prog001	Testovací popis programu	6	Neschváleno	09-02-037-KM 300			Ne
Prog001	Testovací popis programu	6	Neschváleno	03-02-018-KM 200			Ne
Prog001	Testovací popis programu	6	Neschváleno	03-02-006-KM 160			Ne
Prog001	Testovací popis programu	6	Neschváleno	03-02-004-KM 030			Ne
Prog002		8	Neschváleno	03-02-008-KM 050			Ne

CSV XML CSV SQL

Přidat řídicí program

Obrázek 19: Pharis Řídící programy

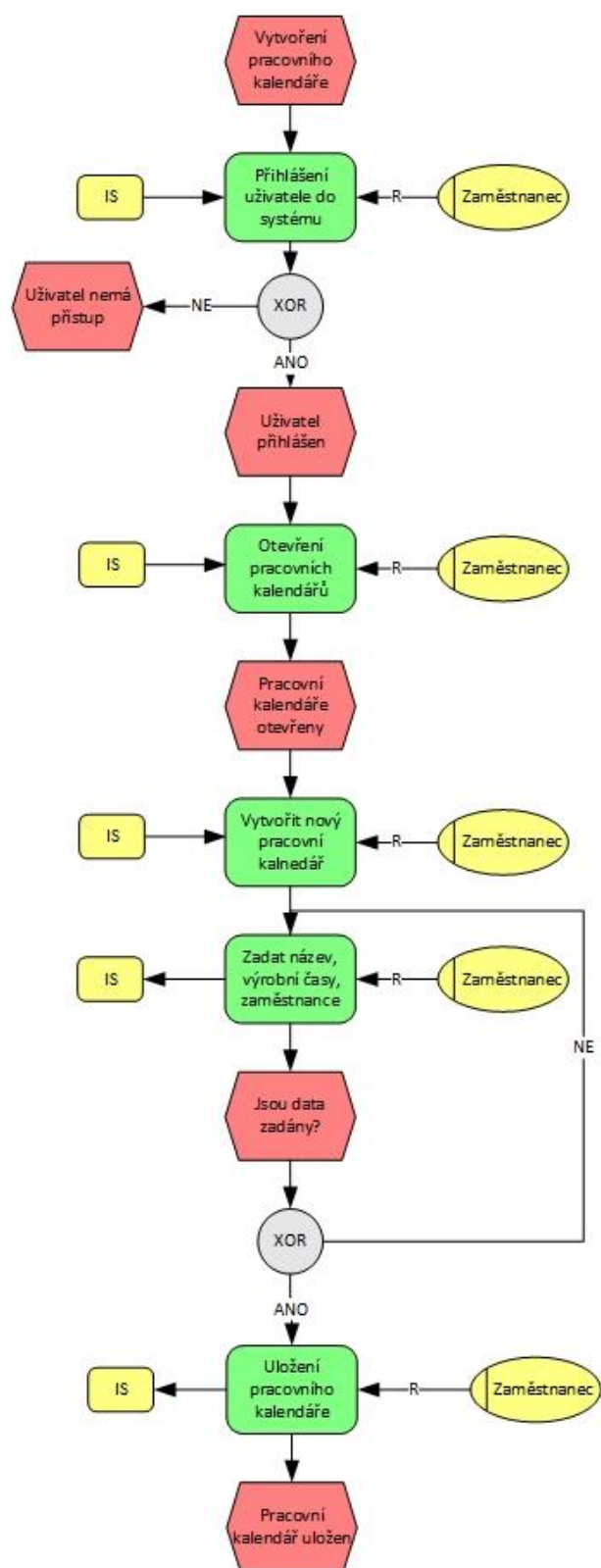
Zdroj: (Vlastní)

3.6 Testování

Při procesu užití pracovních kalendářů se nejprve uživatel přihlásí do systému pomocí svých přístupových údajů, pokud nemá uživatel vytvořen přístup do systému, je mu přístup odepřen. Pokud má uživatel oprávnění a je úspěšně přihlášen, otevře pracovní kalendáře a klikne na tlačítko Nový pracovní kalendář.

V následujícím kroku zadá uživatel do systému jednotlivé časy směn a počet zaměstnanců. Poté proběhne kontrola zadaných údajů, pokud jsou údaje nedostatečné, doplní uživatel potřebné informace. Pokud jsou informace kompletní, uživatel uloží pracovní kalendář stisknutím příslušného tlačítka. Pracovní kalendář je nyní uložen do databáze pracovních kalendářů.

3.6.1 Proces vytvoření pracovního kalendáře



Obrázek 20: EPC diagram vytvoření pracovního kalendáře

Zdroj: (Vlastní)

3.7 Analýza systému

3.7.1 SWOT

Tabulka 1: SWOT analýza

Zdroj: (Vlastní)

Silné stránky	Slabé stránky
<ul style="list-style-type: none">• Eliminace papírové formy ve výrobě• Pokrytí 100% výrobních procesů• Zvýšení efektivnosti výroby• Reporty• Nejprodávanější v ČR	<ul style="list-style-type: none">• Podrobné články ve Pharis Wiki• Detailní testování dle požadavků zákazníka
Příležitosti	Hrozby
<ul style="list-style-type: none">• Rozšíření systému• Expanze na nové trhy• Mobilní zařízení	<ul style="list-style-type: none">• Napadení systému• Zanedbání vývoje systému

Silné stránky:

Informační systém PHARIS je schopen pokrýt veškeré výrobní procesy, dle zjištěných informací systém také výrazně snížil dobu setrvání výrobku na jednotlivých stanicích. Zaměstnanci jednoduše naskenují výrobek a nemusí se zdržovat vedením papírové evidence. Jedná se o neprodávanější výrobní informační systém, a to díky vysoké úrovni řešení servisních požadavků, množství modulů systému a v neposlední řadě také velmi atraktivní ceně. Součástí systému je také možnost využití reportů, tyto reporty si uživatel může nastavit dle svých specifických požadavků. Lze zde také sledovat vytíženost jednotlivých zaměstnanců nebo dobu výroby na jednotlivých stanicích.

Slabé stránky:

MES PHARIS je pokryt řadou testovacích scénářů. Počet testovacích scénářů ale neodpovídá velikosti systému. Tento stav je zapříčiněn stále novými verzemi výrobního informačního systému a velikostí systému PHARIS. Testy jsou vždy vytvářeny tak, aby pokryly nově přidané funkcionality systému. Systém PHARIS poskytuje několik forem uživatelské podpory, mezi ně patří helpdesk, nápověda integrovaná přímo do systému PHARIS, pravidelné školení a nový projekt PHARIS Wiki. Jedná se o webový portál, sloužící jako nápověda k ovládání a používání jednotlivých modulů a funkcionalit informačního systému. I přesto, že portál v současné době obsahuje základní články o modulech systému, je systematicky rozšiřován o další detailnější články.

Příležitosti:

Rozšíření systému o další strategické moduly a funkcionality, dle požadavků zákazníků. Jedním z těchto rozšíření může být například možnost hromadné úpravy zařízení a jejich přiřazení k pracovním kalendářům. Příležitostí je také expanze systému na nové zahraniční trhy. Společnost se se svým systémem zaměřuje převážně na tuzemský trh. V současné době jsou také velmi atraktivní mobilní zařízení a z tohoto důvodu je zde možnost rozšířit PHARIS i na tato zařízení.

Hrozby:

Jako hrozbu lze identifikovat zanedbání systematického vývoje informačního systému. Pokud by byl vývoj systému zanedbán, mohla by nastat situace, ve které by informační systém náhle zůstal pozadu oproti novým trendům a požadavkům trhu a nebyl by tedy atraktivní pro stávající nebo potenciální zákazníky. Jako další hrozba je zde napadení informačního systému. Zabezpečení systému a zálohování dat si zajišťuje každá společnost samostatně, dle doporučení společnosti UNIS. Hrozba odchodu klíčového člena týmu je eliminována díky systému zastupitelnosti.

3.7.2 HOS8

Metoda HOS posuzuje osm klíčových oblastí informačního systému. V případě že jsou jednotlivé části nevyváženy, znamená to neefektivnost celého systému. Metoda HOS posuzuje následující oblasti:

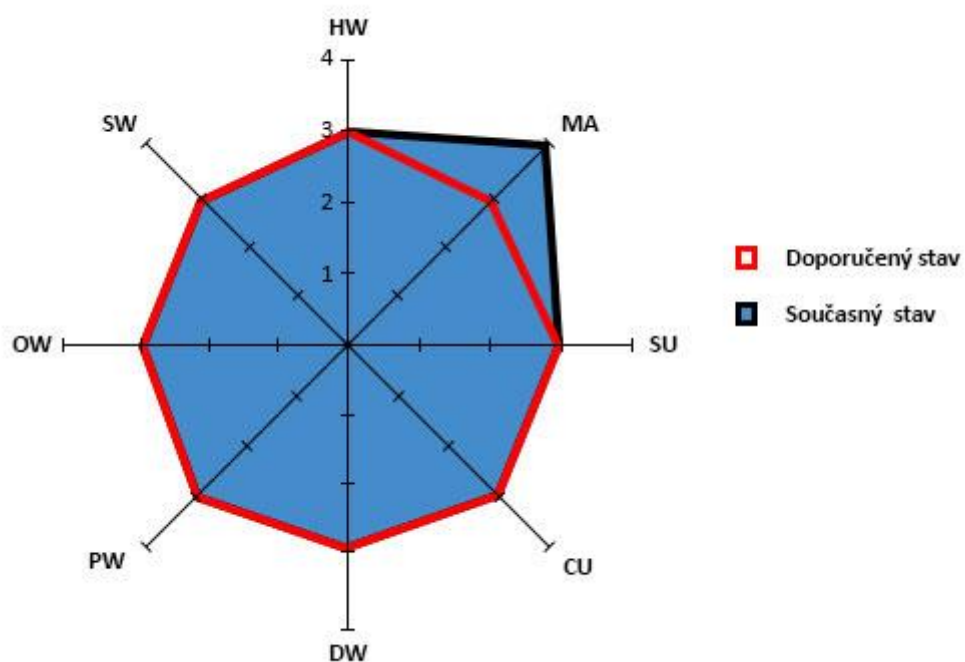
- Hardware
- Software
- Orgware
- Peopleware
- Dataware
- Zákazníci
- Dodavatelé
- Management

Před samotným hodnocením efektivnosti systému je potřeba definovat hodnocení oblastí. Hodnocení je provedeno dle čtyřbodové škály a to následovně:

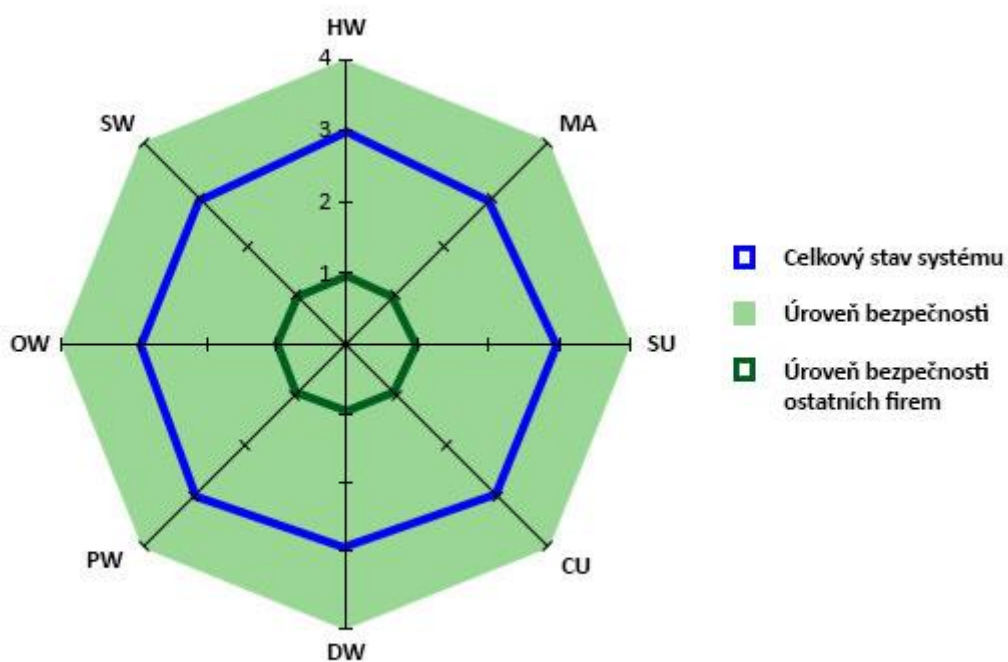
1. Špatná úroveň
2. Spíše špatná úroveň
3. Spíše dobrá úroveň
4. Dobrá úroveň

Z výsledků hodnocení poté rozlišujeme systém vyvážený a nevyvážený. Vyvážený systém je takový, jehož všechny mají stejné hodnocení, nebo se maximálně 3 osy odlišují o hodnotu 1. Nevyvážený systém je takový, který tyto podmínky nesplňuje. Jejich základním problémem je jejich nižší efektivnost. Celková úroveň systému je vždy dána jeho nejslabším článkem.

Při hodnocení informačního systému metodou HOS musíme chápat doporučený stav jako minimální požadovanou úroveň. Ke zlepšení stavu informačního systému je potřeba zaměřit se na dílčí problémy reprezentované jednotlivými otázkami.



Obrázek 21: Graf metody HOS8
Zdroj: (Vlastní)



Obrázek 22: Úroveň bezpečnosti
Zdroj: (Vlastní)

Strategie expanze:

Tuto volbu je velmi důležité zvážit. Zlepšení stavu dané oblasti vyžaduje nemalé finanční prostředky, nesmíme také opomíjet stav oblasti MA a musíme dbát na udržení této úrovně.

Strategie stability:

Ve strategii stability bych doporučil zaměřit se na udržení současného stavu informačního systému.

Strategie omezení:

Strategii omezení je možná pouze v oblasti MA. V této oblasti je možné snížit výdaje.

3.8 Zhodnocení analýzy

Výrobní informační systém PHARIS disponuje vyspělými nástroji a patří mezi nejprodávanější na českém trhu. Pomocí analýzy tohoto informačního systému jsem odhalil drobné nedostatky v oblasti testování nových verzí systému.

Testování je klíčovým faktorem spokojenosti práce s informačním systémem. Je třeba vytvořit standardizovaný formulář pro psaní testovacích scénářů, který v současné situaci ve společnosti chybí. Podklady této diplomové práce byly použity pro automatizované testovací scénáře, které jsou nyní součástí pravidelných automatizovaných testů systému.

Vhodné by bylo kompletovat nedokončené články na webovém portálu PHARIS Wiki, který slouží pro podporu uživatelů systému PHARIS. PHARIS Wiki je jedním z několika možností uživatelské podpory. Uživatelé systému mohou využít HelpDesk, školení ve společnosti nebo nápovědu integrovanou přímo v systému. Ve webovém portálu PHARIS Wiki je potřeba doplnit články, popisující práci s Pracovními kalendáři, Zařízeními, Výrobním modelem, Událostmi a Řídícími programy. Na základě podkladů této diplomové práce jsou postupně doplňovány články na PHARIS Wiki.

4 VLASTNÍ NÁVRHY A ŘEŠENÍ

4.1 Testování systému

Z výsledků analytické části této práce bylo zjištěno, že je nezbytné pokračovat v psaní a tvorbě nových testovacích scénářů a následně tyto scénáře automatizovat. Ve společnosti je úsek testování veden vedoucím zaměstnancem úseku testování. Vedoucí daného úseku je pověřen také řešením servisních požadavků a dalších požadavků. V daném úseku testování je zaměstnáno několik testerů, kteří testují jednotlivé moduly a vymýšlí nové testovací scénáře. Některé scénáře jsou vytvářeny dle požadavků zákazníků. Pro úspěšné testování systému je nezbytná důsledná znalost informačního systému. Proto je nezbytné, aby se každý nový zaměstnanec s ovládáním informačního systému nejprve seznámil. Společnost využívá několik aplikací pro testování. Na těchto aplikacích je spuštěn PHARIS v několika verzích. Na těchto aplikacích se testují nové moduly systému. Pro automatické testování se využívají aplikace označené koncovkou AT.

Plán přípravy a uvolňování jednotlivých verzí systému MES PHARIS

Id	Datum	Akce	Verze	Zbývá dnů		
35	15.08.2020	Implementace	4.08.00	1	Edit	Delete
36	15.08.2020	Implementace	4.08.00	1	Edit	Delete
37	15.08.2020	Implementace	4.08.00	1	Edit	Delete

Podpůrné aplikace a přehledy pro vývoj, servis a implementaci MES PHARIS

Přehled plánovaných implementačních a servisních prací. Task list	Přehled zaplánovaných požadavků do verze 4.08.00 4.08.00	Portál pro nastavení plánu aktualizace testovací aplikace Jenkins	Portál pro zobrazení analýzy a kvality zdrojového kódu DEV Sonar	Přehled nasazených verzí MES PHARIS u jednotlivých zákazníků Release plán
Interní znalostní databáze a řízená dokumentace sekce 300. Wiki 300	Veřejný wiki systém pro zákazníky a partnery systému MES PHARIS MES PHARIS Wiki	Portál se všemi reporty na SQL Server Reporting Services Reporting Services	Slovník nejčastějších a nejvýznamnějších výrazů ve PHARIS. PHARIS Glossary	Statistické přehledy veřejných chyb ve PHARISu. Public bugy

Testovací aplikace MES PHARIS

Testovací aplikace COME DEV. Port terminal serveru 8016. COME DEV	Testovací aplikace COME-MAIN. Port terminal serveru 8010. COME MAIN	Testovací aplikace COME-RELEASE. Port terminal serveru 8019. COME RELEASE	Testovací aplikace pro testování PHARIS IoT. Port terminal serveru 8007. COME DEV IoT	Aut. testování komunikačního rozhraní. Port terminal serveru 8017. COME DEV COMTEST
Aut. test. datových zdrojů (E63, OPC, OPC UA). Port terminal serveru 8002. COME DEV DATASRC	Testovací aplikace pro testování MQB linky. Port terminal serveru 8012. COME DEV MQB	Testovací aplikace pro testování MQB linky. Port terminal serveru 8003. COME MAIN MQB	Automatické testování GUI terminálu. Port terminal serveru 8009. COME DEV TER	Aut. test. zákaznických konfigurací. Port terminal serveru 8008. COME MAIN CUSTCONF
Testovací aplikace COME	Test. aplikace COME MAIN	Test. aplikace pro aut.	aplikace pro prezentaci	aplikace pro prezentaci

Obrázek 23: Servery systému Pharis
Zdroj: (Vlastní)

Příklady aplikací systému PHARIS:

- COME_DEV
- COME_DEV_AT
- COME_MAIN
- COME_MAIN_AT

Testování bylo dokončeno na všech modulech systému PHARIS. Ve společnosti se nyní pracuje na vytváření sofistikovaných testů modulů a tvorbě testovacích scénářů dle zákaznických požadavků. Testování tak rozsáhlého výrobního informačního systému jako je PHARIS, je velmi obtížné. Při testování se dbá na to, aby testovací scénáře pokryly i nečekané vstupy do informačního systému. Testeři jsou zapojeni do procesu testování v plné míře a vytváří testovací scénáře dle předem připraveného harmonogramu. Do procesu testování nejsou zapojeni jen testeři, ale na testování systému se také podílí analytici, kteří provádí částečné manuální testování a vývojáři, kteří se starají o revizi kódu a unit testy.

Moduly informačního systému PHARIS:

- Příprava výroby
- Plán a rozvrh
- Výroba
- Výrobní logistika
- Výrobní zdroje
- Údržba
- Monitoring
- Bilance a ukazatele
- Administrace
- Náповěda

4.1.1 Testovací scénáře

Testovací proces vždy začíná definováním rolí v procesu testování. Vedoucí testování určí rozsah testování a deleguje odpovědnost na testera. Tester se v první řadě seznámí se zadanou formou a rozsahem testování. Po seznámení se zadaným rozsahem testování a důsledným seznámením s testovaným modulem, musí tester vytvořit testovací scénáře.

Testovací scénáře jsou tvořeny tak, aby pokryly většinu simulovaných procesů ve společnosti, odpovídaly zavedeným i nečekaným uživatelským konfiguracím a důkladně prověřily informační výrobní systém PHARIS. Počet testovacích scénářů není v žádném ohledu omezen, dbá se ovšem na to, aby byly pokryty všechny aspekty daného modulu.

Slabinou testovacích scénářů je jejich způsob zadávání. Ve společnosti není stanoven žádný postup ani forma těchto scénářů. Nejvhodnějším řešením by byl návrh šablony pro tvorbu těchto scénářů. Definování jednotné struktury zvýší přehlednost a efektivitu těchto scénářů. Nové testovací scénáře je možné psát v programu Excel, nabízí se také možnost využít komerční nástroje jako Microsoft Test Manager nebo HP Quality Center. Je zde také nezbytná definice výsledků provedení testů. Tento přehled může být využit při kontrole vedoucím pracovníkem. Jednotný grafický styl jasně definuje typ výsledku testu:

Typy výsledků testů:



- Test proveden úspěšně



- Test proveden neúspěšně



- Test nelze provést / nutná konzultace



- Test neproveden

Návrh šablony testovacího scénáře:

Hlavička	Stav testu
Postup testu	
Patička	Autor

Obrázek 24: Wireframe šablony scénáře

Zdroj: (Vlastní)

Hlavička by měla obsahovat jednoznačný identifikátor ID, Název testu, Typ testu, Modul a Kategorie. ID testu musí být jednoznačný identifikátor. Typ testu lze definovat jako manuální nebo automatický, Modul uvádí, ve kterém modulu test probíhá. Do oblasti Stav testu se zadá příslušná barva dle stavu testu. Zelená barva pro úspěšné testování, červená barva pro neúspěšné testování, žlutá barva pro konzultaci s vedoucím a bílá barva indikuje že test ještě nebyl proveden. Postup testu popisuje jednotlivé kroky procesu testování. Tyto kroky jsou naplánovány podle autora testu. V patičce je popsán požadovaný výsledek testu, vyhrazené místo pro případné poznámky a prostor pro autora testu. Poslední pole je určeno pro jméno testera a datum realizace testu.

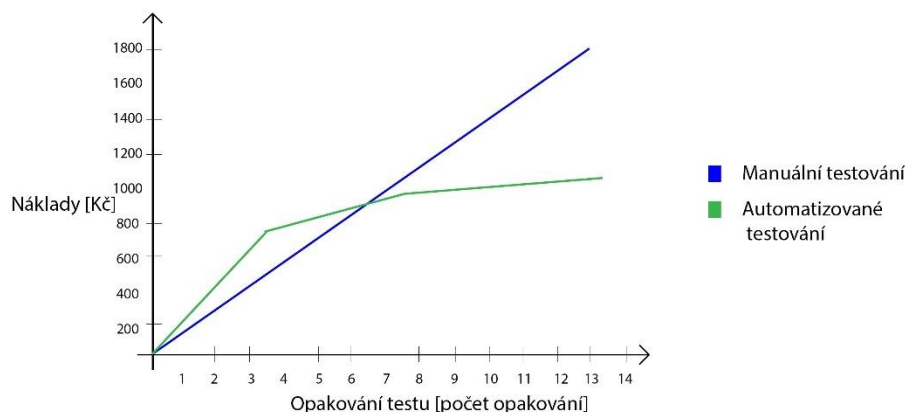
ID:	20170233	
Název:	Nový pracovní kalendář	
Typ:	Automatický test	
Modul:	Pracovní kalendáře	
Postup:	kliknout "Výrobní zdroje"	
	kliknout "pracovní kalendáře" (menu)	
	kliknout "pracovní kalendáře "	
	kliknout "seznam pracovních kalendářů"	
	kliknout "vytvořit nový pracovní kalendář"	
	Vyplnit "Název"	
	Vyplnit " časy "	
	kliknout "Uložit"	
Očekávaný výsledek:	Zápis pracovního kalendáře	
Vstupní podmínky		
Alternativní scénář		
Poznámky		
Test provedl:	3.5.2017, Tester Testovací	Hlaváč Petr 26.04.2017

ID:	20170244	
Název:	Vytvoření typu události	
Typ:	Manuální test	
Modul:	Události	
Postup:	kliknout "Výrobní zdroje"	
	kliknout "události" (menu)	
	kliknout "typy událostí"	
	kliknout "nový typ události"	
	název, popis,	
	událost provoz	
	barva textu bílá	
	podbarvení žluté	
Očekávaný výsledek:	Vytvoření typu události	
Vstupní podmínky		
Alternativní scénář		
Poznámky	Po výběru podbarvení vyskočí dialogové okno a systém nahlásí chybu	
Test provedl:	8.5.2017, Tester Testovací	Hlaváč Petr 28.04.2017

Obrázek 25: Šablony scénářů
Zdroj: (Vlastní)

4.1.2 Manuální testování

Manuální testování je časově i finančně mnohem náročnější než testování automatizované. Testovací scénáře pro manuální testování jsou v převážné většině napsány tak, aby pokryly oblasti výrobního systému, které nepokryjí automatizované testy.



Obrázek 26: Manuální a Automatizované testování

Zdroj: (Vlastní)

Manuální testování je ve společnosti využíváno stejnou měrou jako testování automatizované. Proč tedy manuální testování stále využívat? Manuální testování je velmi efektivní v případech, kdy je potřeba otestovat funkcionality systému, které není možné pokrýt automatizovaným testováním. Mezi tyto funkcionality můžeme zařadit například samostatnou kategorii souborových příloh.

Souborové přílohy se dají k těmto zařízením připojit jak při vytvoření nového zařízení, tak při následné editaci již vytvořeného zařízení. Přílohy mohou být nahrány ve formátech CSV, PDF nebo EXCEL. Souborové přílohy většinou obsahují technické nákresy nebo technologické specifikace dané formy nebo zařízení. Přílohy se dají připojit dvěma způsoby. Jedním z nich je připojení přílohy z místa v síti a druhým je nahrání souborové přílohy z počítače. Tester vyvolá pomocí tlačítka dialogové okno, v tomto dialogovém okně se zobrazí lokální umístění souborů.

Tester k zařízení vyhledá příslušný soubor, který vybere a následně přiřadí k danému stroji. Tímto způsobem je souborová příloha nahrána k danému zařízení. Příloha a zařízení jsou spárovány. Při běžném provozu si tak může zaměstnanec zobrazit veškeré technické popisy k danému zařízení. Souborové přílohy lze v průběhu využití stroje editovat a aktualizovat, přílohy jde také smazat.

4.1.3 Automatizované testování

Automatizované testování je v dnešní době velmi rozšířený způsob testování. Ve společnosti je využíváno jak manuální, tak automatizované testování. Oproti manuálnímu testování jsou automatické testy mnohem rychlejší, efektivnější, přesnější a ekonomicky výhodnější.

Při automatizovaném testování je prvním krokem napsání testovacího scénáře. Obrovskou výhodou tohoto typu testování je fakt, že testy jsou spouštěny automaticky, bez přítomnosti testerů. Testy jsou realizovány pomocí skriptů, které jsou psány v programovacím jazyku C#.

Tento objektově orientovaný programovací jazyk vyvinula firma Microsoft a je založen na programovacích jazycích C++ a Java. Ve společnosti je vytvořen framework, který slouží jako odrazový můstek pro testery. Tento framework byl vytvořen vedoucím pracovníkem oddělení testování. Testovací skripty jsou vytvářeny v programu Microsoft Visual Studio.

Framework obsahuje základní stavební prvky testování. Framework obsahuje základní příkazy, které se využívají při vytváření testovacích metod. Tester tedy nemusí tvořit nové testovací podmínky. Tyto podmínky jsou již vytvořeny a tester je tedy může kdykoliv využívat nebo si je přizpůsobit pro potřebu specifického testování.

```
/// <summary>
/// Metoda otestuje zda se objevil Message Box
/// </summary>
/// <param name="testName">název aktuálního testu</param>
public void InfoMessageBox(string testName)
{
    if (GHP.SeleniumIsElementPresent("PhoenixDialogBox101", "imgMsgBox", "id"))
    {
        string ExceptionMessage = GHP.SeleniumGetText(GHP.FindId("PhoenixDialogBox101_ctl01_lblMsgBox"));
        CloseDialogWindow();
        TC.OutputBlue(testName + " : InfoMessageBox: " + GHP.RemoveAccents(ExceptionMessage));
    }
}
/// <summary>
/// Metoda ověří existenci položky ve stromě
/// </summary>
/// <param name="itemName">název položky ve stromě</param>
/// <returns></returns>
public bool CheckDeleteItemInTree(string itemName)
{
    GHP.SeleniumType("id=TreeViewSearchBox", itemName);
    Thread.Sleep(100);
    GHP.SeleniumKeyPress("id=TreeViewSearchBox"); //send enter
    Thread.Sleep(300);
    if (GHP.SeleniumIsElementPresent("link=" + itemName, "", "xxx") == true)
        return false;
    else
        return true;
}
/// <summary>
/// Metoda vybere položku ve stromě
/// </summary>
/// <param name="itemName">název položky ve stromě</param>
public void SearchInTree(string itemName)
{
    GHP.SeleniumType("id=TreeViewSearchBox", itemName);
    Thread.Sleep(1000);
    GHP.SeleniumKeyPress("id=TreeViewSearchBox"); //send enter
    Thread.Sleep(3500);
    GHP.SeleniumClick(String.Format("/a[contains(text(),'{0}')]\"", itemName));
}
/// <summary>
/// Metoda vybere položku ve stromě pro dialogové okno
/// </summary>
/// <param name="itemName">název položky ve stromě</param>
```

Obrázek 27: Framework v C#

Zdroj: (Vlastní)

Při automatizovaném testování se využívají testovací scénáře a testovací metody:

```
//Pracovní kalendare

[TestMethod]
public void ADDCalendar() // Pracovní kalendář
//přidání kalendáře, kontrola zapsání, přidání směn a smazání kalendáře
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow],
    Configuration.VarInt[Configuration.CSVDataRow]);
    PD.ControlWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.Editcalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void ADDExceptCalendar()
// přidání kalendáře, kontrola zapsání, přidání směn a smazání kalendáře Vyjíměčný kalendář
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow],
    Configuration.VarInt[Configuration.CSVDataRow]);
    PD.ControlWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.AddExceptCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.Editcalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void ADDSpecialCalendar()
// přidání kalendáře(+přidání vyjimky) a odebrání vyjíměčného kalendáře
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow],
    Configuration.VarInt[Configuration.CSVDataRow]);
    PD.AddException(Configuration.VarUniqueString[Configuration.CSVDataRow],
    Configuration.VarUniqueString[Configuration.CSVDataRow] + "R");
    PD.AddspecialCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void AddExpecta()
// přidání vyjimky, ověření vytvoření a smazání vyjimky (tydenní)
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddExcept(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.CheckExcept(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.Delexcept(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void AddExpectb()
// přidání vyjimky a smazání vyjimky (měsíční)
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddExceptMonth(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.Delexcept(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}
```

Obrázek 29: Scénáře v C#

Zdroj: (Vlastní)

```

//Zařízení

    public void AddNewEquipment(string text, string longText, string number)
    //Případ: Přidání nového zařízení.
    {
OpenDeviceManagementModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
        GHP.ClickById("btnAddButton");
        GHP.TypeById("ebName", text);
        GHP.TypeById("ebIdentificationNo", DateTime.Now.ToString("yyyyMMddHHmm"));
        GHP.TypeById("ebDescription", longText);
        GHP.TypeByName("ebEvNumber", "652");
        GHP.TypeByName("ebSerialNo", number);
        GHP.TypeById("ebDeviceLocation", text);
        GHP.TypeById("EBDate", DateTime.Now.ToString("dd.MM.yyyy"));
        GHP.TypeById("ebWorkingLifeCycle", number);
        GHP.TypeById("ebWorkingLifeInStandardHours", number);
        GHP.ClickByIdNowait("cbxHasMonitor");
        GHP.ClickByIdNowait("bxUseForPlanningUI");
        GHP.TypeById("ebOperationRegistrationCount", number);
        GHP.TypeById("ebAllOperationsRegistrationCount", number);
        GHP.TypeById("ebCountProducts", number);
        GHP.ClickById("btnSaveButton");

    }

    public void ControlEquipment(string text)
    //kontrola vytvoření nového zařízení
    {
OpenDeviceManagementModule(System.Reflection.MethodBase.GetCurrentMethod().ToString()
);
        PC.SetFilter("MasterTwinPanel_phoenixContent_devUDevices_dtgDevices", "1", text,
false);
    }

    public void DeleteEquipment(string text)
    //smazání zařízení
    {
OpenDeviceManagementModule(System.Reflection.MethodBase.GetCurrentMethod().ToString()
);
        PC.SetFilterAndOpenFirstItemFromDg("dtgDevices", "1", text, false);
        GHP.ClickById("MasterTwinPanel_phoenixContent_devUDeviceProfile_btnDeleteButton");
        GHP.ClickById("btnYES");
    }

    public void AddEquipmentType(string text)
    //přidání typu zařízení
    {
OpenDeviceManagementModule(System.Reflection.MethodBase.GetCurrentMethod().ToString()
);

        GHP.ClickByName("ctl00$MasterTwinPanel$phoenixContent$MasterTabBar$TabButtonDeviceTyp
es$ctl00");
        GHP.ClickById("MasterTwinPanel_phoenixContent_devUDeviceTypes_btnAddButton");
        GHP.TypeById("ebDeviceTypeName", text);
        GHP.SelectById("ddlDeviceTypeSubType", "pq-option-0-6");
        GHP.TypeById("ebDeviceTypeDescription", "Testovací typ");
        GHP.ClickById("btnSaveButton");
    }

    public void ControlAddEquipmentType(string text)
    //kontrola přidání typu zařízení
    {
OpenDeviceManagementModule(System.Reflection.MethodBase.GetCurrentMethod().ToString()
);
        GHP.ClickByName("ctl00$MasterTwinPanel$phoenixContent$MasterTabBar$TabButtonDeviceTyp
es$ctl00");
        PC.SetFilter("dtgDeviceTypes", "1", text, false);
    }

```

Obrázek 30: Vytvoření zařízení v C#

Zdroj: (Vlastní)

Scénář a metoda

```
//Zařízení
[TestMethod]
public void AddEquipment() // Zařízení
                           //přidání zařízení, kontrola zapsání a smazání zařízení
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddNewEquipment(Configuration.VarUniqueString[Configuration.CSVDataRow],
        Configuration.VarString10[Configuration.CSVDataRow],
        Configuration.VarInt[Configuration.CSVDataRow]);
    PD.ControlEquipment(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteEquipment(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    Thread.Sleep(1000);
    SU.Logout();
}
```



4.1.4 Reportování nalezených chyb při testování

Nedostatky v programu můžeme rozdělit na dvě kategorie: Chybu (Bug) a Vlastnost (Feature). Chyba (Bug) je stav, kdy funkcionálita nefunguje dle požadavku a musí se opravit. Vlastnost (Feature) je funkcionálita realizovaná dle požadavku, ale není uživatelsky přívětivá, proto se na ni musí napsat nový požadavek na úpravu. Pokud se při testování funkcionality nalezne chyba, tak se nalezený problém nahlásí do systému pomocí tiketu. K tiketu se také přiřadí název, identifikace verze a aplikace, oblast, kde byla chyba nalezena a také na koho byl tiket odeslán. Vytvořenému tiketu se poté přidělí priorita. Dle zadané priority je poté tiket předán na příslušné oddělení, kde je tiket vyřešen.

Mezi tyto chyby můžeme zařadit například špatné propojení databáze zařízení nebo pracovních skupin v modulu Pracovní kalendáře. V případě že nefunguje správné propojení s databází, nezobrazí se v tomto modulu seznam vytvořených zařízení nebo pracovních skupin. Pokud se tento seznam nezobrazí, nelze přiřadit pracovní kalendář k požadovanému zařízení nebo pracovní skupině.

Chyba systému (Bug)

Jako příklad chyby lze uvést proces vytvoření výjimky při vytváření nového pracovního kalendáře. Při vytváření nového pracovního kalendáře se pomocí tlačítka přidat výjimku načte nová stránka, na které je možné novou výjimku vytvořit. Pokud se ale vybere týdenní, měsíční nebo roční opakování výjimky, nezobrazí se dodatečné checkboxy a další možnosti potřebné pro nastavení opakování výjimky.

Detail výjimky

Pracovní čas			
Od	Do	Využitelnost	Množství personálu
09:00	13:00	04:00	1
13:00	19:00	00:00	0

Opakování výjimky

☐ Denní Každý 2 den

☒ Týdenní

☐ Měsíční

☐ Roční

Platnost výjimky

Začíná 10.3.2017

☒ Ukončit po: 3 opakování

☐ Ukončit do: 15.3.2017

Obrázek 31: Detail Výjimky
Zdroj: (Vlastní)

Tento problém byl klasicky nahlášen pomocí tiketu a v rámci několika hodin byla chyba opravena.

Obrázek 32: Opakování výjimky

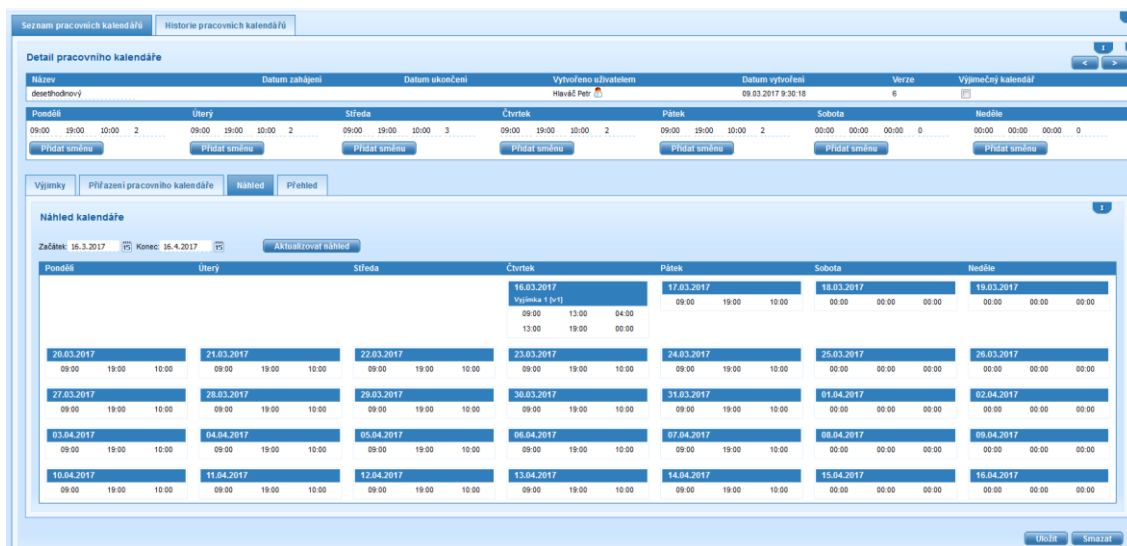
Zdroj: (Vlastní)

Vlastnost systému (Feature)

Uživatel si také může povšimnout informační ikony umístěné v pravém horním rohu, tato ikona slouží k zobrazení uživatelských šablon. Pokud ale uživatelská šablona neexistuje, zobrazí se dialogové okno, které obsahuje informace o souboru stránky a souboru šablony. Jedná se o vlastnost systému, pro její úpravu se musí napsat požadavek. Toto okno slouží také k identifikaci stránky a jejímu umístění na serveru. Zobrazuje také informaci, zda existuje uživatelská šablona stránky. Pro nedostatečně proškoleného pracovníka může být tato informace matoucí a bylo by vhodné dialogové okno doplnit odkazem na nápovědu nebo přímo na tvorbu uživatelských šablon.

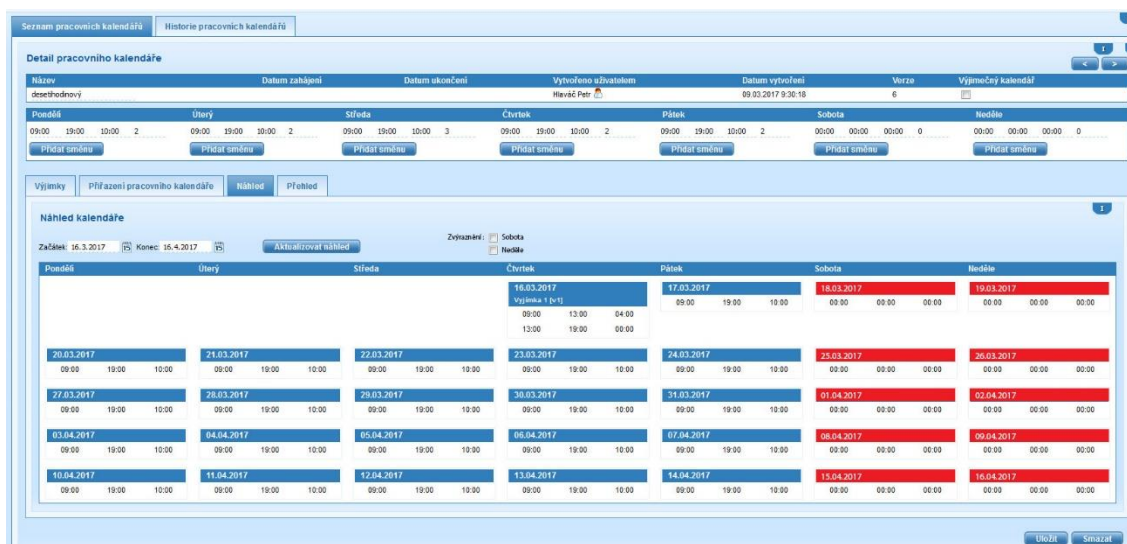
Obrázek 33: Soubor šablony

Zdroj: (Vlastní)

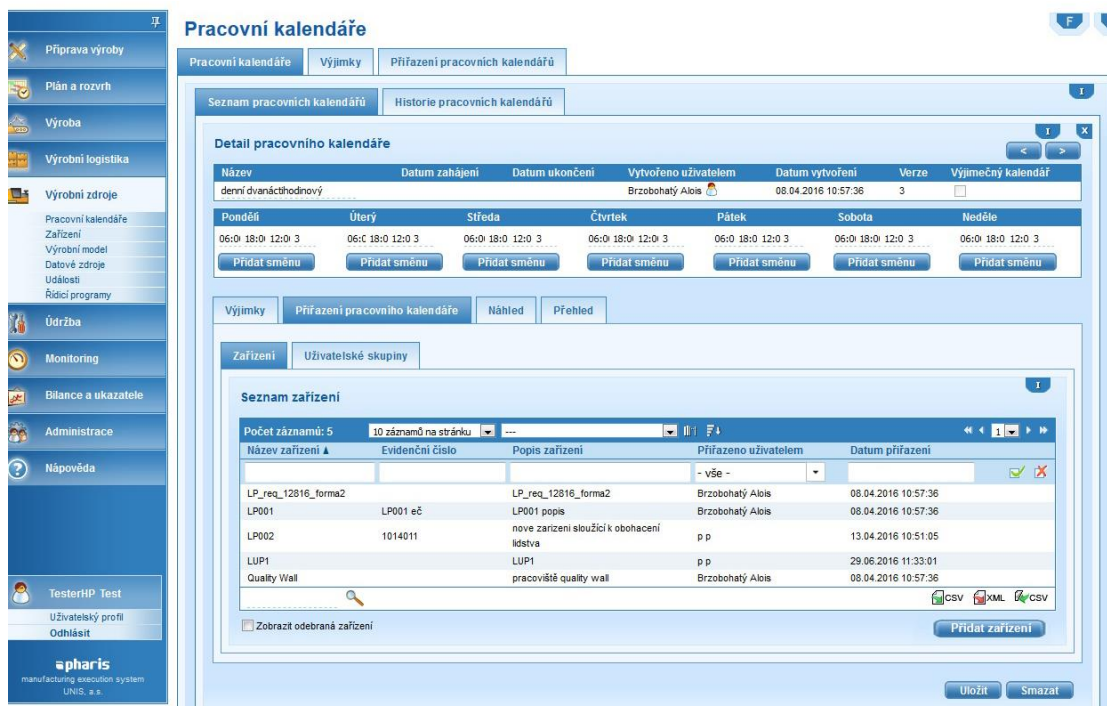


Obrázek 34: Seznam pracovních kalendářů
Zdroj: (Vlastní)

Při zobrazení náhledu pracovního kalendáře nelze efektivně odlišit víkendy od pracovních dnů. Z tohoto důvodu je vhodné, aby byla pro podniky možnost zvýraznit tyto nepracovní dny, pro lepší přehlednost. Jedná se o vlastnost systému a pro její úpravu musí být vytvořen požadavek na vytvoření checkboxů. Řešení spočívá ve vytvoření checkboxů Sobota a Neděle, které budou umístěny za tlačítkem Aktualizovat náhled. Po kliknutí na příslušný checkbox se v přehledu zobrazí barevně odlišené nepracovní dny. Tento krok slouží pro lepší přehlednost.



Obrázek 35: Návrh seznamu pracovních kalendářů
Zdroj: (Vlastní)



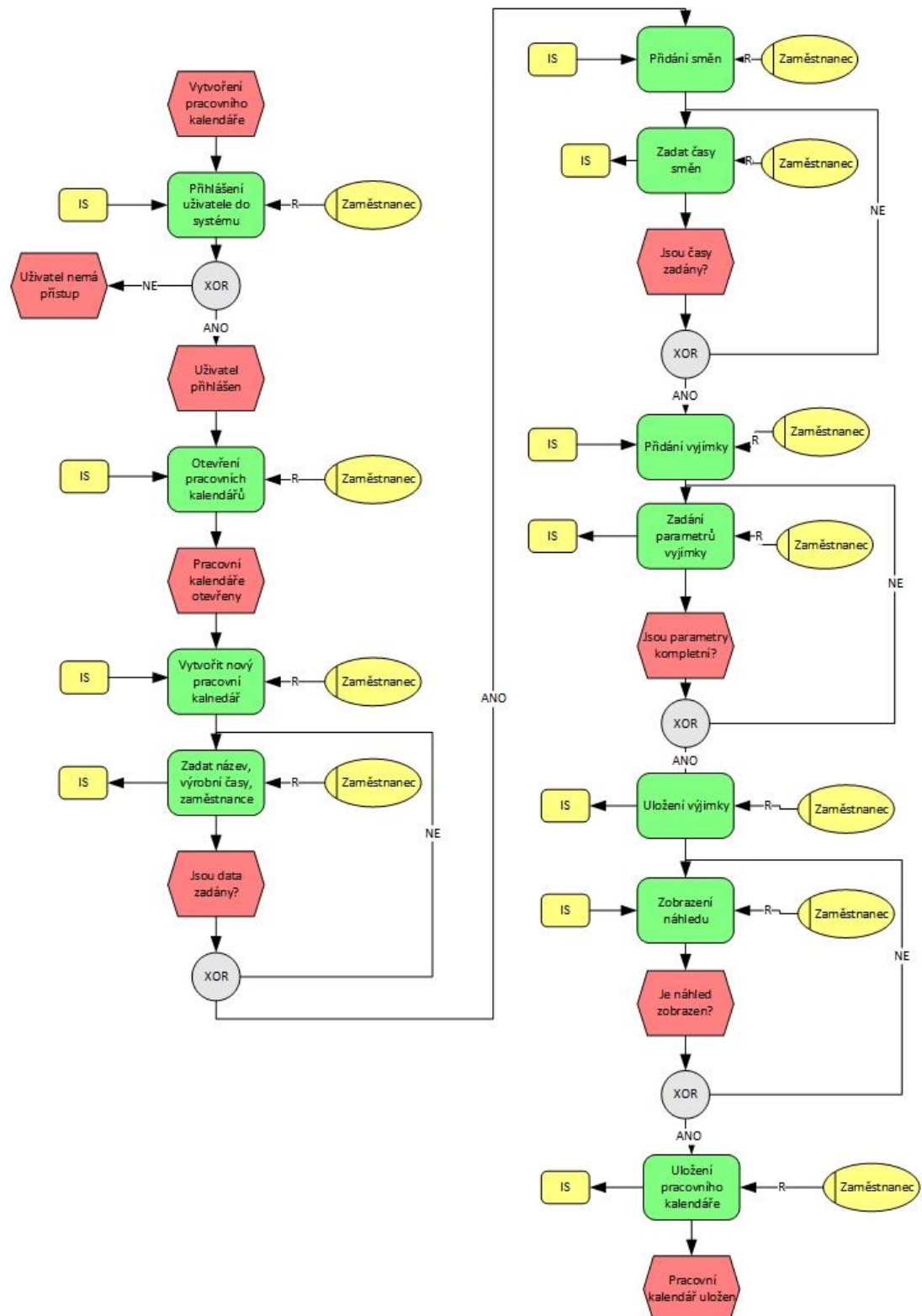
Obrázek 36: Přřazení zařízení
Zdroj: (Vlastní)

Jedná se o vlastnost systému a pro její úpravu se musí napsat požadavek na přidání tlačítka Přidat zařízení. Po stisku tohoto tlačítka se zobrazí seznam aktuálních zařízení v dialogovém okně. Seznam by měl obsahovat sloupce: Checkbox, Název zařízení, Evidenční číslo, Typ zařízení, Typová příbuznost, Umístění, Popis zařízení a Zahrnout do plánování. Díky přidání checkboxů před Názvy zařízení je uživatel schopen vybrat větší počet požadovaných zařízení a přiřadit tato zařízení k pracovnímu kalendáři. V současnosti musí uživatel přiřazovat zařízení k pracovnímu kalendáři jednotlivě.

Počet záznamů: 2173 10 záznamů na stránku						
Název zařízení ▲	Evidenční číslo	Typ zařízení	Typová příbuznost	Umístění	Popis zařízení	Zahrnout do plánování
<input type="checkbox"/>		- vše -	- vše -	- vše -		<input type="checkbox"/>
<input type="checkbox"/> 00305		Stroj	Stroj		Hloubička Innovation	Ne
<input type="checkbox"/> 00JS Forma		Forma	Forma			Ne
<input type="checkbox"/> 111	111	Stroj	Stroj	g1	Vop	Ne
<input type="checkbox"/> 111222333		Stroj	Stroj			Ano
<input type="checkbox"/> 111a		Stroj	Stroj		Vop	Ne
<input type="checkbox"/> 112	tak a je to	Stroj	Stroj	P13	112	Ano
<input type="checkbox"/> 113	113	Stroj	Stroj	113	113	Ano
<input type="checkbox"/> 2678 udrtest		Přípravek	Přípravek		udrtest	Ne
<input type="checkbox"/> 3505105LR		Stroj	Stroj		Kontrolní stanice Flansch CONTI(L359)	Ne
<input type="checkbox"/> 53453	123	Forma	Forma	123	ghxg	Ne

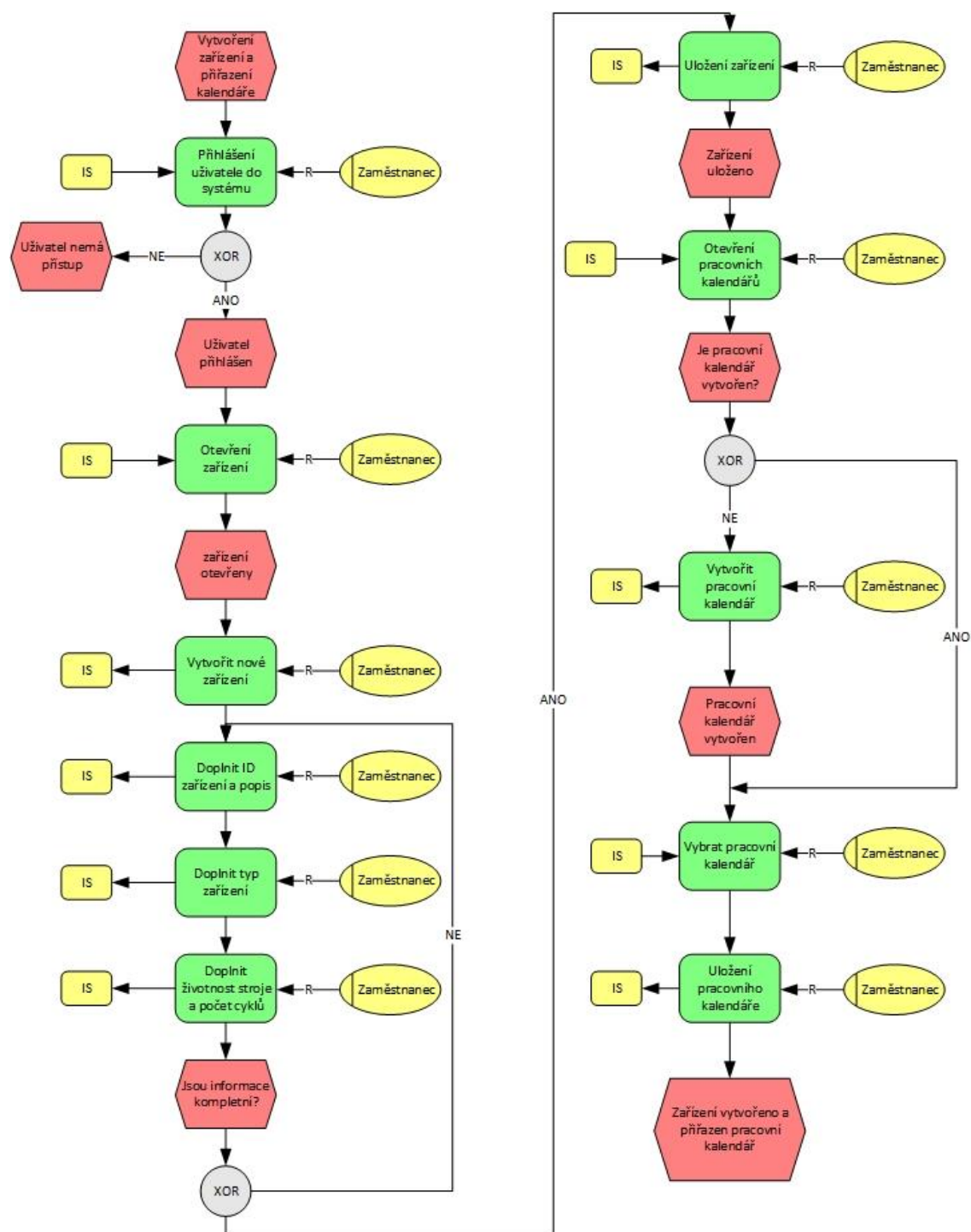
Obrázek 37: Seznam zařízení
Zdroj: (Vlastní)

Proces vytvoření pracovního kalendáře



Obrázek 38: Vytvoření pracovního kalendáře
Zdroj: (Vlastní)

Proces vytvoření zařízení typu stroj a přiřazení kalendáře



Obrázek 39: Proces vytvoření zařízení

Zdroj: (Vlastní)

Proces vytvoření pracovního kalendáře

Proces vytvoření pracovního kalendáře začíná přihlášením uživatele do informačního výrobního systému PHARIS. Uživatel na přihlašovací stránce zadá přihlašovací údaje, pokud jsou přihlašovací údaje nesprávné, může uživatel zadat své údaje znovu. Uživatel musí mít přidělena patřičná oprávnění. V případě že uživatel nemá vytvořen účet v systému, je požadavek na přihlášení odmítnut, protože uživatel nemá přístup do systému. Po úspěšném přihlášení do systému, otevře zaměstnanec modul Výrobní zdroje a zde otevře Pracovní kalendáře. Po otevření pracovních kalendářů, klikne zaměstnanec na tlačítko Vytvořit nový pracovní kalendář. Na stránce vytvoření nového pracovního kalendáře zadá uživatel časy směn, počet zaměstnanců a denní počet hodin. Zaměstnanec nyní zkontroluje, zda jsou data zadány, pokud ne, doplní chybějící údaje, pokud ano, pokračuje uživatel k přidání směn. Uživatel zadá časy směn, pokud nejsou kompletní, údaje doplní. V okamžiku kdy jsou doplněny údaje, přistoupí zaměstnanec k vytvoření výjimky. Po zadání parametrů výjimky uživatel zobrazí náhled pracovního kalendáře, pokud náhled souhlasí s požadovaným pracovním kalendářem, uživatel pracovní kalendář uloží do systému.

Proces vytvoření zařízení a přiřazení kalendáře

Při procesu vytvoření zařízení a následného přiřazení pracovního kalendáře se uživatel nejprve přihlásí do systému pomocí svého přihlašovacího jména a hesla. Pokud uživatel nemá vytvořen účet, je mu přístup do systému odepřen. Po zadání platné kombinace a hesla je uživatel přihlášen do systému PHARIS. Uživatel musí mít přidělena patřičná oprávnění. V dalším kroku otevře uživatel modul Výrobní zdroje a zde otevře Zařízení. V zařízeních uživatel přidá nové zařízení pomocí tlačítka Přidat nové zařízení. Po zobrazení nabídky tvorby nového zařízení vyplní uživatel potřebné údaje, mezi které patří ID, název, popis, typ zařízení, výrobní číslo, evidenční číslo, životnost a jednotky. Pokud jsou údaje kompletní, uloží uživatel nové zařízení do databáze. Nyní je zařízení uloženo. Uživatel otevře pracovní kalendáře a vybere pracovní kalendář, pokud není vytvořen tak pracovní kalendář vytvoří. Vytvořený kalendář uživatel uloží do databáze pracovních kalendářů. V tento okamžik uživatel potvrdí přiřazení pracovního kalendáře k zařízení a uloží pracovní kalendář. V tuto chvíli je zařízení vytvořeno a je k němu přiřazen pracovní kalendář.

4.2 PHARIS Wiki

Informační výrobní systém PHARIS poskytuje profesionální servisní podporu. Po implementaci PHARISU je organizováno školení pro ovládání informačního systému. Toto školení probíhá pro klíčové i běžné uživatele výrobního systému.

Informační systém PHARIS je dodáván s manuálem na používání systému. Pokud ale uživatel potřebuje efektivně získat informace o dané oblasti systému, není nic jednoduššího než využít rychlou nápovědu pomocí portálu PHARIS Wiki.

PHARIS Wiki je nejnovější projekt k poskytování podpory zákazníkům. Mimo tento portál PHARIS disponuje také manuálem na používání systému, helpdeskem a školením zaměstnanců. Portál PHARIS Wiki slouží jako pomocný portál pro informační výrobní systém. Portál je zabezpečen pomocí přihlášení kvalifikovaného uživatele. Uživatel při přihlášení zadá uživatelské jméno a heslo. V případě že je kombinace uživatelského jména a hesla nesprávná, je uživateli odmítnut přístup. Pokud je kombinace správná, je uživateli umožněn přístup do portálu.

PHARIS Wiki popisuje moduly informačního systému a podrobný popis práce s jednotlivými částmi tohoto systému. PHARIS Wiki je kategorizován do určitých kategorií a specifických podkategorií. Názvy kategorií odpovídají názvům jednotlivým modulům informačního systému PHARIS. Mezi výše uvedené kategorie patří Příprava výroby, Plán a rozvrh, Výroba, Výrobní logistika, Výrobní zdroje, Údržba, Monitoring, Bilance a ukazatele a Administrace. Po výběru dané kategorie se zobrazí příslušné podkategorie. V kategorii Výrobní zdroje jsou obsaženy tyto podkategorie: Pracovní kalendáře, Zařízení, Model výroby, Události a Řídící programy.

Portál obsahuje základní informace o jednotlivých modulech systému. Pro prezentaci tohoto portálu je tedy potřeba doplnit informace a grafické materiály k jednotlivým kategoriím i podkategoriím. V případě, že budou tyto informace doplněny, bude portál poskytovat uživatelskou podporu na velmi vysoké úrovni.

V současné době probíhá pravidelné a systémové doplňování článků PHARIS Wiki. V následujících kategoriích bude navrženo doplnění informací pro kategorii Výrobní zdroje.

4.2.1 Pracovní kalendáře

Tato podkategorie je dokončena ve značné míře. Uvedené informace jsou podány vyčerpávajícím způsobem, podávají dostatečný popis jednotlivých funkcí a částí této podkategorie. Zpracování postupu práce seznámí uživatele detailně s postupy při práci s pracovními kalendáři.

Díky informacím uvedeným v této podkategorii je uživatel schopen vytvořit pracovní kalendář, vytvořit výjimečné kalendáře, vytvořit výjimky, přiřadit kalendáře k jednotlivým strojům, editovat již vytvořené kalendáře nebo zobrazit náhled těchto kalendářů.

4.2.2 Zařízení

Podkategorie Zařízení obsahuje základní informace. Články jsou aktuální a současně vzniká nová koncepce článků. Oblast Zařízení by měla čtenáře seznámit s obecnou charakteristikou Zařízení a jejím propojení s ostatními systémy.

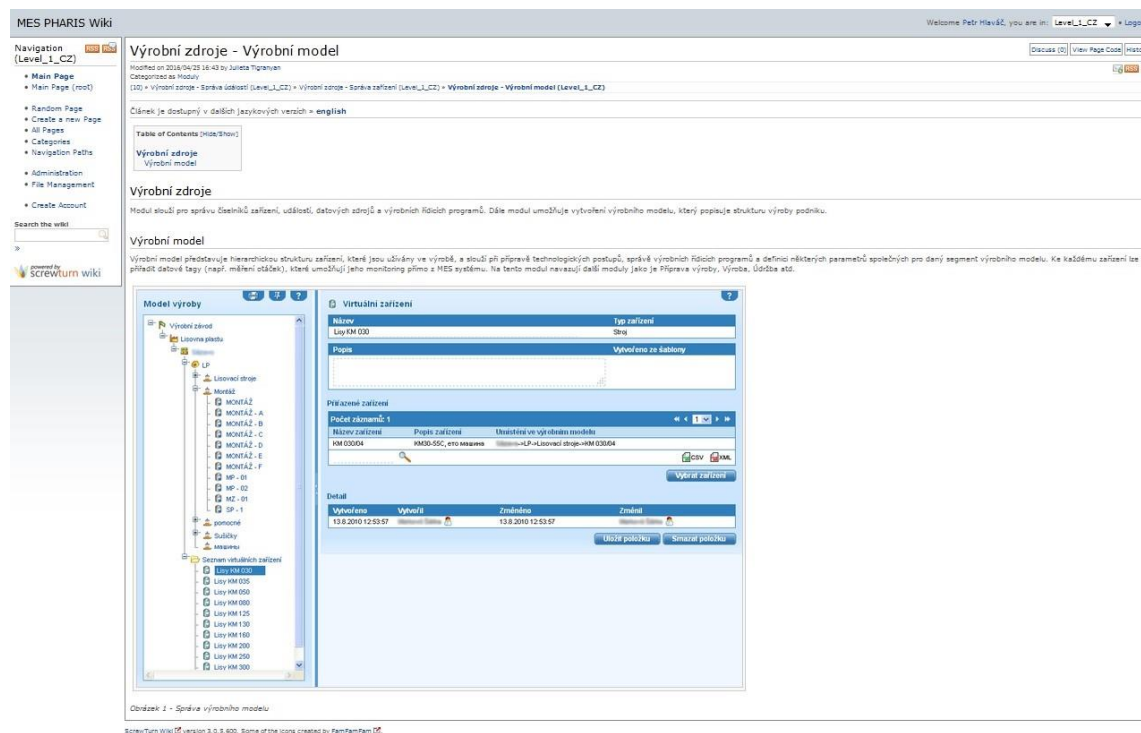
Po uvedení čtenáře do tématu by měl následovat popis jednotlivých částí modulu. Dle přiloženého textu by měl být uživatel schopen vytvořit nové zařízení nebo přiřadit k zařízení souborové přílohy a také by měl být schopen doplnit nezbytné atributy jako je identifikační číslo, umístění zařízení, RFID kód, počet souběžných operací a v neposlední řadě také životnost stroje. Uživatel by měl být také schopen definovat o jaký typ zařízení se jedná.



Obrázek 40: Wiki – zařízení
Zdroj: (Vlastní)

4.2.3 Výrobní model

Podkapitola Výrobní model by měla popisovat strukturu výroby podniku. Současná verze není kompletní, ale probíhá pravidelné doplňování článků. Po seznámení uživatele s výrobním modelem, by zde měly být definovány jednotlivé části výrobního modelu. Dále by zde měly být definovány postupy při vytváření těchto jednotlivých částí. V modelu výroby může uživatel vytvářet strukturu podniku pomocí tvorby oblastí, provozoven, výrobních jednotek nebo terminálů. Vytvářené zařízení se řadí do stromové struktury.



Obrázek 41: Wiki - Výrobní model
Zdroj: (Vlastní)

4.2.4 Události

V podkategorii Správa událostí je definován princip fungování správce událostí. V článku by měl být popsán také postup přidání nové události, tyto události lze následně přidat k jednotlivým strojům. Článek by měl také obsahovat popis vytvoření nového typu události.

4.3 Mobilní aplikace

Mobilní aplikace je vhodný směr pro rozšíření portfolia stávajících klientů informačního výrobního systému PHARIS. Mobilní zařízení jsou využívány ve velké míře a pro manažery společnosti je velmi výhodné mít po ruce nástroj, který dokáže zobrazit data v reálném čase. Vytvořením mobilní aplikace mohou manažeři systému v kterýkoliv okamžik sledovat aktuální výrobu ve společnosti, kontrolovat výrobní listy nebo zobrazit vytvořené reporty.

Aplikace by měla být vytvořena pro mobilní systémy Android. Později i pro IOS.



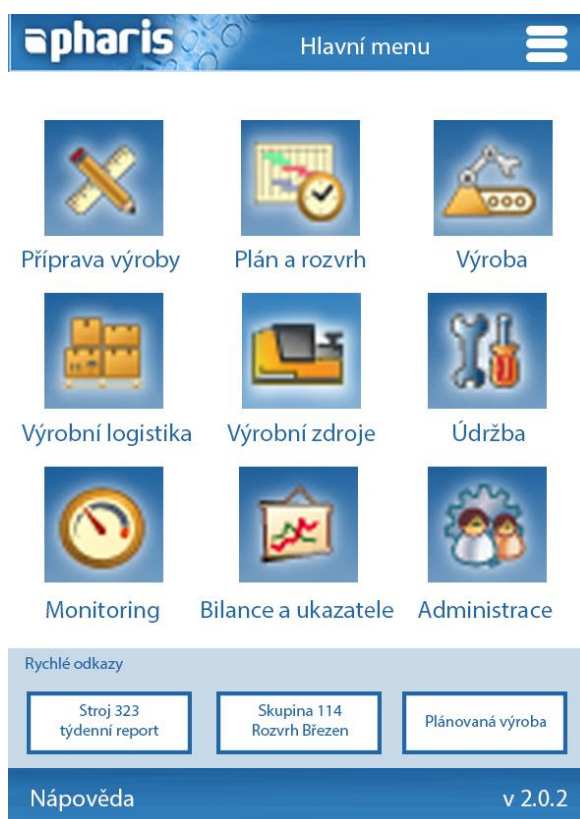
Obrázek 44: Wireframe mobilní aplikace
Zdroj: (Vlastní)

Aplikace je rozdělena na čtyři základní části. V hlavičce aplikace je umístěno logo výrobního informačního systému, umístění, ve kterém se uživatel v daný okamžik nachází a tlačítko rychlé volby. Toto tlačítko slouží k rychlé navigaci po informačním systému a po jeho stisknutí se zobrazí jednotlivé moduly informačního systému.

Následující část je nejrozsáhlejší a slouží pro zobrazení základních modulů jako jsou:

- Příprava výroby
- Plán a rozvrh
- Výroba
- Výrobní logistika
- Údržba
- Monitoring
- Balance a ukazatele
- Administrace
- Nápověda

Spodní část obrazovky slouží jako prostor pro umístění rychlých odkazů. Uživatel se díky těmto odkazům dostane okamžitě na potřebnou část systému. Okamžitě si tak může zobrazit například poslední reporty. V patičce návrhu jsou umístěny informace o verzi aplikace.



Obrázek 45: Návrh mobilní aplikace

Zdroj: (Vlastní)

pharis

OEE

Zařízení:

Dev1

Začátek:

4.6.2017 14:05:22

Uložit

Konec:

12.6.2017 11:07:24

Zařízení: Dev1

Období: 4.6.2017 14:05:22 - 12.6.2017 11:07:24



Analýza ztrát		
Organizační ztráta	<div></div>	0,76%
Ztráty prostoje	<div></div>	44,63%
Ztracený výkon	<div></div>	4,32%
Ztráta kvality	<div></div>	3,15%

Nápověda

v 2.0.2

pharis

OEE - časová osa

Zařízení:

Dev1

Začátek:

4.6.2017 14:05:22

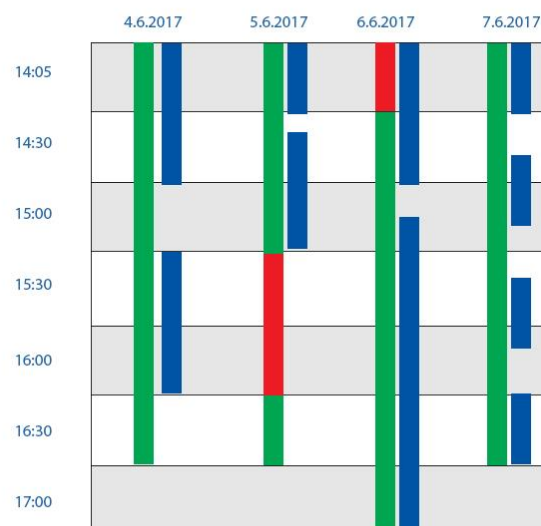
Uložit

Konec:

12.6.2017 11:07:24

Zařízení: Dev1

Období: 4.6.2017 14:05:22 - 12.6.2017 11:07:24



Nápověda

v 2.0.2

Obrázek 46: Návrhy OEE v mobilní aplikaci

Zdroj: (Vlastní)

4.4 Ekonomické zhodnocení

Tato část diplomové práce je zaměřena na vyčíslení potřebných nákladů, které jsou nezbytné při realizaci navržených řešení. Následně definuji přínosy, které realizace změn přinese.

4.4.1 Náklady a přínosy

V této kapitole vyčísím náklady, které jsou potřebné pro realizaci navržených změn. Společnost s investicemi do systému PHARIS počítá a očekává také přínosy po realizaci navržených změn. Následující tabulka zobrazuje kalkulace jednotlivých navržených řešení. Tyto náklady se mohou zvýšit během realizace navržených řešení, s tímto faktem musí společnost také kalkulovat.

Tabulka 2: Náklady

Zdroj: (Vlastní)

Popis	Cena
Testování modulu Výrobní zdroje	60 000,- Kč
Mobilní aplikace	140 000,- Kč
Celkem	200 000,- Kč

Částky uvedené v tabulce byly vypočítány vynásobením průměrné hodinové sazby a počtu odpracovaných hodin. Jedná se o pevné náklady na prvotní dokončení. Během realizace navržených změn lze předpokládat vyšší výdaje, než jsou uvedeny v tabulce, proto je velmi důležité provádět pravidelně revizi těchto činností. Kalkulovaná cena mobilní aplikace je cena za jednu aplikaci na jednu konkrétní platformu. Aplikace by měla sloužit k zobrazení výrobních zdrojů. Uvedené kalkulace slouží pouze jako odhad výsledné ceny.

4.4.2 Přínosy

Přínosy by měly být pro společnost atraktivní, jedná se o velmi důležitý faktor, podle kterého se vedení společnosti bude rozhodovat, zda rozhodne o realizaci navržených změn.

Po realizaci navržených změn můžeme očekávat následující přínosy:

- Efektivnější testování modulu Výrobní zdroje
- Eliminace nalezených chyb před vydáním nové verze
- Efektivnější testování systému PHARIS
- Definování jednotné struktury testovacích scénářů
- Doplnění článků webového portálu PHARIS Wiki
- Rozšíření systému na mobilní zařízení

Tabulka 3: Přínosy

Zdroj: (Vlastní)

Popis	Cena
Úspora času testerů, analytiků a vývojářů	240 000,- Kč
Celkem	240 000,- Kč

Částka uvedená v tabulce byla vypočítána vynásobením počtu ušetřených hodin, průměrné hodinové sazby a počtu zainteresovaných zaměstnanců. Uvedená částka slouží jako přibližné vyčíslení a jedná se o odhad.

Navržené změny mohou znamenat eliminaci chyb při práci s informačním výrobním systémem PHARIS. Systém bude atraktivnější pro potenciální zákazníky, díky spokojenosti stávajících zákazníků a jejich kladným referencím. Jednotná struktura formuláře pro psaní testovacích scénářů přinese úsporu času. Po dokončení článků na webovém portálu PHARIS Wiki bude portál sloužit jako plnohodnotný prvek nápovědy.

ZÁVĚR

Cílem mé diplomové práce bylo posouzení informačního výrobního systému PHARIS společnosti UNIS a.s., a následný návrh změn. Informační systém podle společnosti obsahuje několik nedostatků. Tyto nedostatky byly odhaleny pomocí analýzy informačního systému a následně bylo navrženo vhodné řešení pro eliminaci těchto nedostatků.

V teoretické části diplomové práce jsou popsány základní pojmy informačních systémů. Základní problémy výrobního informačního systému PHARIS byly odhaleny při analýze tohoto informačního systému. Informační systém byl analyzován pomocí metody HOS 8 a pomocí metody SWOT. Výsledkem těchto analýz byla identifikace nejčastějších problémů se systémem. Testování tak rozsáhlého systému jako je PHARIS je velmi obtížné, systém je v současné situaci pokryt testováním, neustále se ale pracuje na propracovanějších testovacích případech. Portál PHARIS Wiki je také neustále doplňován novými články, podpora systému je ale stále na profesionální úrovni, a to díky manuálu systému, helpdesku či sofistikovanému systému školení.

Na základě zjištěných výsledků analýz bylo navrženo řešení současného stavu výrobního informačního systému. Bylo potřeba definovat jednotný styl tvorby testovacích scénářů. Následně jsem se zaměřil na tvorbu automatizovaných testů pro modul Výrobní zdroje. Vytvořil jsem skripty pro základní testování tohoto modulu.

Po zavedení těchto testů byly odhaleny chyby v systému, jejichž řešení je popsáno v následující části. Jsou zde také navrženy dva rozšířené procesy uživatelských postupů pro oblast zařízení a oblast pracovních kalendářů. V následující části je popsán webový portál PHARIS Wiki, který slouží jako nápověda pro zákazníky a uživatele systému. V této části je představen návrh na aktualizaci článků v modelu Výrobní zdroje. Detailnější informace o jednotlivých částech systému, mohou znamenat efektivnější práci se systémem.

V další části je navržen možný vzhled a funkcionalita rozšíření výrobního informačního systému na mobilní zařízení. Následně byly definovány náklady a přínosy navržených změn. Realizace navržených změn by měla odstranit chyby v modulu Výrobní zdroje, rozšířit webový portál o podrobnější informace a rozšířit informační systém PHARIS i na mobilní zařízení.

SEZNAM POUŽITÉ LITERATURY

- (1) SKLENÁK, Vilém a kol. *Data, informace, znalosti a Internet*. Praha: C. H. Beck, 2001. 507 s. ISBN 80-7179-409-0.
- (2) HRONEK, Jiří. *Informační systémy* [online]. Přírodovědecká fakulta Univerzita Palackého, Katedra informatiky: 2007 [cit. 2017-01-12]. Dostupné z: <http://phoenix.inf.upol.cz/esf/ucebni/infoSys.pdf>
- (3) GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. *Podniková informatika -2. přepracované a aktualizované vydání*. Praha: Grada publishing a.s., 2009. 496 s. ISBN 978-80-247-2615-1
- (4) TVRDÍKOVÁ, Milena. *Aplikace moderních informačních technologií v řízení firmy, Nástroj ke zvyšování kvality informačních systémů*. Praha: Grada Publishing, 2008. 173 s. ISBN 978-80-247-2728-8.
- (5) BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: Podnik v informační společnosti – 3., aktualizované a doplněné vydání*. Praha: Grada Publishing, a.s., 2012. 323 s. ISBN 978-80-247-7594-4.
- (6) SODOMKA, Petr a Hana KLČOVÁ. *Informační systémy v podnikové praxi*. Brno: Computer Press, 2010. 501 s. ISBN 978-80-251-2878-7.
- (7) POUR, Jan. *Informační systémy a technologie*. Praha: Vysoká škola ekonomie a managementu, 2006. 492 s. ISBN 80-86730-03-4.
- (8) ŘEPA, Václav. *Analýza a návrh informačních systémů*. Praha: Ekopress, 1999. 403 s. ISBN 80-861-1913-0.
- (9) VYMĚTAL, Dominik. *Informační systémy v podnicích: teorie a praxe projektování*. Praha: Grada, 2009. 142 s. ISBN 978-80-247-3046-2.

- (10) Co je MES: Mescentrum. Úvod [online]. Dostupné z: <http://mescentrum.cz/o-projektu/co-mes>
- (11) SCHWALBE, Kathy. *Řízení projektů v IT*. Brno: Computer Press, 2007. 720 s. ISBN 978-80-251-1526-8.
- (12) ŠTĚDRŮ, Bohumír. *Open Source software ve veřejné správě a soukromém sektoru*. 1. vyd. Praha: Grada, 2009, s. 57. ISBN 978-80-247-3047-9.
- (13) LACKO, Ľuboslav. *1001 tipů a triků pro SQL*. Brno: Computer Press, 2011. 416s. ISBN 978-80-251-3010-0.
- (14) CONOLLY, Thomas, Carolyn E BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. 488 s. ISBN 978-80-251-2328-7.
- (15) KOCH, Miloš a Bernard Neuwirth. *Datové a funkční modelování*. Vyd. 4., rozšířené. Brno: Akademické nakladatelství CERM, 2010. 142 s. ISBN 978-80-214-4125-5.
- (16) GÁLA, Libor, Jan POUR a Prokop TOMAN. *Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi, technologie informačních systémů, řízení a rozvoj podnikové informatiky*. 1. vyd. Praha: Grada, 2006, s. 332. ISBN 80-247-1278-4.
- (17) PARSONS, David. *Dynamic Web Application Development Using XML and Java*. Singapore: Cengage Learning EMEA, 2008. ISBN 97-818-448-0541-9.
- (18) ONDRÁK, Viktor, Petr SEDLÁK a Vladimír MAZÁLEK. *Problematika ISMS v manažerské informatice*. Vyd. 1. Brno: CERM, 2013. ISBN 978-80-7204-872-4.

- (19) MLÝNEK, Jaroslav. 2007. *Zabezpečení obchodních informací*. Vyd. 1. Brno: Computer Press. ISBN 978-80-251-1511-4.
- (20) ROOT.CZ. 2015. Kybernetická bezpečnost: O čem je nový zákon? *ROOT.CZ* [online]. [cit. 2017-02-13]. Dostupné z: <http://www.ROOT.CZ/clanky/kyberneticka-bezpecnost-o-cem-je-novy-zakon/>
- (21) MANAGEMENTMANIA.COM. 2016. Outsourcing. *MANAGEMENTMANIA.COM* [online]. [cit. 2017-02-18]. Dostupné z: <https://MANAGEMENTMANIA.COM/cs/outsourcing>
- (22) MANAGEMENTMANIA.COM. 2015. Cloud Computing. *MANAGEMENTMANIA.COM* [online]. [cit. 2017-02-19]. Dostupné z: <https://MANAGEMENTMANIA.COM/cs/cloud-computing>
- (23) PROGRAMUJTE. Odborný web zaměřený na oblast vývoje desktopových aplikací. *Programujte.cz* [online]. ©2014 [cit. 2017-03-02]. Dostupné z: <http://programujte.com/clanek/2013050100-tvorba-reportu-v-ms-reporting-services-2008-1-dil/>
- (24) PEAKPOINTNET.CZ. 2010. Softwarový proces z pohledu zákazníka - Informační systém. *PEAKPOINTNET.CZ* [online]. [cit. 2017-03-01]. Dostupné z: <http://www.PEAKPOINTNET.CZ/cs/nabizime/softwarovy-proces/informacni-system>
- (25) KNEŠL, Jiří. 2009. Agilní vývoj: Úvod. *Zdrojak.cz* [online]. [cit. 2017-03-01]. Dostupné z: <https://www.zdrojak.cz/clanky/agilni-vyvoj-uvod/>
- (26) Veřejný rejstřík a Sběrka listin - Ministerstvo spravedlnosti České republiky. [online]. Copyright © 2012 [cit. 2017-02-22]. Dostupné z: <https://or.justice.cz/ias/ui/rejstrik-firma.vysledky?subjektId=202551&typ=PLATNY>

- (27) UNIS, a.s.. Regionální hospodářská komora Brno [online]. Copyright © 1996 [cit. 2017-03-04]. Dostupné z: <http://www.rhkbrno.cz/en/catalogue-of-members/1DCB0151-43A8-408A-91AE-CA5AA5355D35>
- (28) MES PHARIS®, Výrobní informační systém. MES PHARIS®, Výrobní informační systém [online]. [cit. 2017-04-11]. Dostupné z: <http://www.pharis.cz/>
- (29) MES PHARIS® - O systému PHARIS®. MES PHARIS®, Výrobní informační systém [online]. [cit. 2017-04-21]. Dostupné z: <http://www.pharis.cz/cs/o-systemu>
- (30) Veřejný rejstřík a Sbírka listin - Ministerstvo spravedlnosti České republiky. [online]. Copyright © 2012 [cit. 2017-04-14]. Dostupné z: <https://or.justice.cz/ias/ui/vypis-sl-detail?dokument=44793100&subjektId=202551&spis=686917>
- (31) PATTON, Ron. 2002. *Testování softwaru*. Praha: Computer Press. ISBN 80-7226-636-5.

SEZNAM OBRÁZKŮ

Obrázek 1: Kategorie podnikových dat	14
Obrázek 2: Řetězec informačního systému	15
Obrázek 3: Základní vymezení IS/ICT	16
Obrázek 4: Rozdělení informačních systémů	18
Obrázek 5: Kategorie informačních systémů.....	20
Obrázek 6: Webová aplikace	26
Obrázek 7: Proces zabezpečení IS	28
Obrázek 8: Zvážení rizika a opatření	29
Obrázek 9: Cloud computing	30
Obrázek 10: Logo společnosti	34
Obrázek 11: Organizační schéma společnosti	37
Obrázek 12: Struktura společnosti	38
Obrázek 13: ERP a Pharis.....	41
Obrázek 14: Klienti systému Pharis.....	42
Obrázek 15: Pharis výrobní kalendáře	44
Obrázek 16: Pharis Správa zařízení	45
Obrázek 17: Pharis Výrobní model	46
Obrázek 18: Pharis Události	47
Obrázek 19: Pharis Řídící programy	48
Obrázek 20:EPC diagram vytvoření pracovního kalendáře	49
Obrázek 21: Graf metody HOS8	53
Obrázek 22: Úroveň bezpečnosti	53
Obrázek 23: Servery systému Pharis	55
Obrázek 24: Wireframe šablony scénáře	58
Obrázek 25: Šablony scénářů	59
Obrázek 26: Manuální a Automatizované testování	60
Obrázek 27: Framework v C#.....	61
Obrázek 28: Metody v C#.....	62
Obrázek 29: Scénáře v C#	62
Obrázek 30: Vytvoření zařízení v C#	63
Obrázek 31: Detail Výjimky	65

Obrázek 32: Opakování výjimky	66
Obrázek 33: Soubor šablony	66
Obrázek 34: Seznam pracovních kalendářů	67
Obrázek 35: Návrh seznamu pracovních kalendářů	67
Obrázek 36: Přiřazení zařízení	68
Obrázek 37: Seznam zařízení	68
Obrázek 38: Vytvoření pracovního kalendáře	69
Obrázek 39: Proces vytvoření zařízení	70
Obrázek 40: Wiki – zařízení	73
Obrázek 41: Wiki - Výrobní model	74
Obrázek 42: Wiki – Události	75
Obrázek 43: Wiki - Řídící programy	75
Obrázek 44: Wireframe mobilní aplikace	76
Obrázek 45: Návrh mobilní aplikace	77
Obrázek 46: Návrhy OEE v mobilní aplikaci	78

SEZNAM GRAFŮ

Graf 1: Vývoj obratu (v mil. Kč)	39
Graf 2: Vývoj počtu zaměstnanců	39
Graf 3: Podíl zakázek a dotované projekty	40

SEZNAM TABULEK

Tabulka 1: SWOT analýza	50
Tabulka 2: Náklady	79
Tabulka 3: Přínosy	80

SEZNAM PŘÍLOH

Příloha 1: Testovací scénáře	I
Příloha 2: Testovací metody	IV

Příloha 1: Testovací scénáře

//Pracovní kalendare

```
[TestMethod]
public void ADDCalendar() // Pracovní kalendář
//přidání kalendáře, kontrola zapsání, přidání směn a smazání kalendáře
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow],
        Configuration.VarInt[Configuration.CSVDataRow]);
    PD.ControlWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.AddExceptCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void ADDExceptCalendar()
// přidání kalendáře, kontrola zapsání, přidání směn a smazání kalendáře Vyjimečný kalendář
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow],
        Configuration.VarInt[Configuration.CSVDataRow]);
    PD.ControlWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.AddExceptCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.Editcalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void ADDSpecialCalendar()
// přidání kalendáře(+přidání vyjímky) a odebrání vyjimečného kalendáře
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow],
        Configuration.VarInt[Configuration.CSVDataRow]);
    PD.AddException(Configuration.VarUniqueString[Configuration.CSVDataRow],
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "R");
    PD.AddspecialCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void AddExpecta()
// přidání vyjímky, ověření vytvoření a smazání vyjímky (tydenní)
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddExcept(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.CheckExcept(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DelExcept(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void AddExpectb()
// přidání vyjímky a smazání vyjímky (měsíční)
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddExceptMonth(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DelExcept(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}
```

```

[TestMethod]
public void AddExpectc()
// přidání vyjímky a smazání vyjímky (roční)
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddExceptYear(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DelExcept(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void ADDandEditSpecialCalendar()
// přidání vyjíměčného kalendáře a přidání směn, odebrání
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow],
        Configuration.VarInt[Configuration.CSVDataRow]);
    PD.AddspecialCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.Editcalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void AssignCalendar()
// přiřazení pracovního kalendáře ke stroji a smazání přidání
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AssigmentWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteAssigmentWorkCalendar(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

//Zařízení

[TestMethod]
public void AddEquipment() // Zařízení
//přidání zařízení, kontrola zapsání a smazání zařízení
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddNewEquipment(Configuration.VarUniqueString[Configuration.CSVDataRow],
        Configuration.VarString10[Configuration.CSVDataRow],
        Configuration.VarInt[Configuration.CSVDataRow]);
    PD.ControlEquipment(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteEquipment(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    Thread.Sleep(1000);
    SU.Logout();
}

[TestMethod]
public void DelEquipment()
//odebrání zařízení
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.DeleteEquipment(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
public void AddTypeEquipment()
//přidání typu zařízení a kontrola a smazání
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddEquipmentType(Configuration.VarUniqueString[Configuration.CSVDataRow]);

    PD.DeleteEquipmentType(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    Thread.Sleep(1000);
    SU.Logout();
}

```

```

// řídicí programy
[TestMethod]
public void AddPnpcProgram()
// přidání řídicího programu, kontrola, přiřazení zařízení, schválení, přidání
// komentáře, reakce na komentář a smazání
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.ADDPcncPrograms(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.ControlPcncPrograms(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.AddDeviceToPcnc(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.AddCommentPcncPrograms(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.ReactCommentPcncPrograms(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.ApprovePcncPrograms(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeletePcncPrograms(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    Thread.Sleep(1000);
    SU.Logout();
}

// události

[TestMethod]
public void AddEvent()
// přidání, kontrola, přidání periodické události a smazání události k zařízení
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PD.AddEvent(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.CheckEvent(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.AddPeriod(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    PD.DeleteEvent(Configuration.VarUniqueString[Configuration.CSVDataRow]);
    SU.Logout();
}

[TestMethod]
// přidání nové oblasti, provozovny, centra, jednotky, zařízení a terminálu a
// postupné smazání
public void AddAreas()
{
    SU.Login(System.Reflection.MethodBase.GetCurrentMethod().ToString());

    PD.OpenProductionManagementModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PC.CreateNodeToProdModel("Site", "",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "S");
    PC.CreateNodeToProdModel("Area",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "S",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "A");
    PC.CreateNodeToProdModel("ProductionCenter",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "A",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "PC");
    PC.CreateNodeToProdModel("Unit",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "PC",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "U");
    PC.CreateNodeToProdModel("Equipment",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "U",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "E", "STROJHPP");
    PC.CreateNodeToProdModel("Terminal",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "PC",
        Configuration.VarUniqueString[Configuration.CSVDataRow] + "T", "FOR0MPC012");
    PD.DeleteArea(Configuration.VarUniqueString[Configuration.CSVDataRow] + "T");
    PD.DeleteArea(Configuration.VarUniqueString[Configuration.CSVDataRow] + "E");
    PD.DeleteArea(Configuration.VarUniqueString[Configuration.CSVDataRow] + "U");
    PD.DeleteArea(Configuration.VarUniqueString[Configuration.CSVDataRow] + "PC");
    PD.DeleteArea(Configuration.VarUniqueString[Configuration.CSVDataRow] + "A");
    PD.DeleteArea(Configuration.VarUniqueString[Configuration.CSVDataRow] + "S");
    SU.Logout();
}

```

Příloha 2: Testovací metody

```
//VÝROBNÍ ZDROJE
//Pracovní kalendáře

//(Pracovní kalendáře)
public void AddWorkCalendar(string text, string number)
// Přidání nového zařízení.
{
    // string a = DateTime.Now.AddHours(7).ToString("HHmm"); //kontrolní proměnná
    datetime

    OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    GHP.ClickById("btnNewWorkCalendar");
    GHP.TypeById("ebTitle", text);
    GHP.TypeById("ebTimeFrom1Day1", "08:00");
    GHP.TypeById("ebTimeTo1Day1", "16:00");
    GHP.TypeById("ebEfficiency1Day1", "08:00");
    GHP.TypeById("ebTheNumberOfStaff1Day1", "2");

    GHP.TypeById("ebTimeFrom1Day2", "08:00");
    GHP.TypeById("ebTimeTo1Day2", "16:00");
    GHP.TypeById("ebEfficiency1Day2", "08:00");
    GHP.TypeById("ebTheNumberOfStaff1Day2", "2");

    GHP.TypeById("ebTimeFrom1Day3", "08:00");
    GHP.TypeById("ebTimeTo1Day3", "16:00");
    GHP.TypeById("ebEfficiency1Day3", "08:00");
    GHP.TypeById("ebTheNumberOfStaff1Day3", "2");

    GHP.TypeById("ebTimeFrom1Day4", "08:00");
    GHP.TypeById("ebTimeTo1Day4", "16:00");
    GHP.TypeById("ebEfficiency1Day4", "08:00");
    GHP.TypeById("ebTheNumberOfStaff1Day4", "2");

    GHP.TypeById("ebTimeFrom1Day5", "08:00");
    GHP.TypeById("ebTimeTo1Day5", "16:00");
    GHP.TypeById("ebEfficiency1Day5", "08:00");
    GHP.TypeById("ebTheNumberOfStaff1Day5", "2");

    GHP.TypeById("ebTimeFrom1Day6", "08:00");
    GHP.TypeById("ebTimeTo1Day6", "16:00");
    GHP.TypeById("ebEfficiency1Day6", "08:00");
    GHP.TypeById("ebTheNumberOfStaff1Day6", "2");

    GHP.TypeById("ebTimeFrom1Day0", "08:00");
    GHP.TypeById("ebTimeTo1Day0", "16:00");
    GHP.TypeById("ebEfficiency1Day0", "08:00");
    GHP.TypeById("ebTheNumberOfStaff1Day0", "2");

    GHP.ClickById("btnSave");
    GHP.ClickById("btnYES");
}

public void AddExceptCalendar(string text)
// přidání vyjímčného kalendáře
{
    OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PC.SetFilterAndOpenFirstItemFromDg("dtgWorkCalendars", "1", text, false);
    GHP.ClickById("chkbIsExceptionable");
    GHP.TypeById("dtpStartTime_EBDate", "12.10.2017");
    GHP.TypeById("dtpStartTime_EBTime", "08:00:00");
    GHP.TypeById("dtpEndTime_EBDate", "28.12.2017");
    GHP.TypeById("dtpEndTime_EBTime", "16:00:00");
    GHP.ClickById("btnSave");
    GHP.ClickById("btnYES");
}
```

```

public void Editcalendar(string text)
//přidání směn
{
    OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PC.SetFilterAndOpenFirstItemFromDg("dtgWorkCalendars", "1", text, false);

    GHP.ClickById("btnAddNewShiftDay1");
    GHP.TypeById("ebTimeFrom2Day1", "16:00");
    GHP.TypeById("ebTimeTo2Day1", "20:00");
    GHP.TypeById("ebEfficiency2Day1", "04:00");
    GHP.TypeById("ebTheNumberOfStaff2Day1", "3");

    GHP.ClickById("btnAddNewShiftDay2");
    GHP.TypeById("ebTimeFrom2Day2", "16:00");
    GHP.TypeById("ebTimeTo2Day2", "20:00");
    GHP.TypeById("ebEfficiency2Day2", "04:00");
    GHP.TypeById("ebTheNumberOfStaff2Day2", "3");

    GHP.ClickById("btnAddNewShiftDay3");
    GHP.TypeById("ebTimeFrom2Day3", "16:00");
    GHP.TypeById("ebTimeTo2Day3", "20:00");
    GHP.TypeById("ebEfficiency2Day3", "04:00");
    GHP.TypeById("ebTheNumberOfStaff2Day3", "4");

    GHP.ClickById("btnAddNewShiftDay4");
    GHP.TypeById("ebTimeFrom2Day4", "16:00");
    GHP.TypeById("ebTimeTo2Day4", "20:00");
    GHP.TypeById("ebEfficiency2Day4", "04:00");
    GHP.TypeById("ebTheNumberOfStaff2Day4", "4");

    GHP.ClickById("btnAddNewShiftDay5");
    GHP.TypeById("ebTimeFrom2Day5", "16:00");
    GHP.TypeById("ebTimeTo2Day5", "20:00");
    GHP.TypeById("ebEfficiency2Day5", "04:00");
    GHP.TypeById("ebTheNumberOfStaff2Day5", "4");

    GHP.ClickById("btnAddNewShiftDay6");
    GHP.TypeById("ebTimeFrom2Day6", "17:36");
    GHP.TypeById("ebTimeTo2Day6", "20:36");
    GHP.TypeById("ebEfficiency2Day6", "03:00");
    GHP.TypeById("ebTheNumberOfStaff2Day6", "2");

    GHP.ClickById("btnAddNewShiftDay0");
    GHP.TypeById("ebTimeFrom2Day0", "18:00");
    GHP.TypeById("ebTimeTo2Day0", "20:00");
    GHP.TypeById("ebEfficiency2Day0", "02:00");
    GHP.TypeById("ebTheNumberOfStaff2Day0", "1");

    GHP.ClickById("btnSave");
    GHP.ClickById("btnYES");
}

public void AddspecialCalendar(string text)
//přidání vyjimečného kalendáře
{
    OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PC.SetFilterAndOpenFirstItemFromDg("dtgWorkCalendars", "1", text, false);
    GHP.ClickById("chkbIsExceptionable");
    GHP.TypeById("dtpStartTime_EBDate", "16.4.2017");
    GHP.TypeById("dtpStartTime_EBTime", "10:00:00");
    GHP.TypeById("dtpEndTime_EBDate", "26.4.2017");
    GHP.TypeById("dtpEndTime_EBTime", "17:00:00");
    GHP.ClickById("btnSave");
    GHP.ClickById("btnYES");
}

```



```

public void AddException(string text, string requestname)
//přidání vyjimky (přes pracovní kalendář)
{
    OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PC.SetFilterAndOpenFirstItemFromDg("dtgWorkCalendars", "1", text, false);
    GHP.ClickById("btnNewWorkCalendarException");
    GHP.TypeById("ctl05_PhoenixDialogBox101_ctl01_workCalendarException_ebName",
    requestname);
    GHP.TypeById("ebTimeFrom1", "18:00");
    GHP.TypeById("ebTimeTo1", "20:00");
    GHP.TypeById("ebEfficiency1", "02:00");
    GHP.TypeById("ebTheNumberOfStaff1", "2");
    GHP.TypeById("ebDailyEveryDay", "2");
    GHP.TypeById("dtpRecurrenceRangeStart_EBDate", "23.6.2017");
    GHP.TypeById("ebEndAfterFixedOccurrences", "24");
    GHP.ClickById("ctl05_PhoenixDialogBox101_ctl01_workCalendarException_btnSave");
    GHP.ClickById("MasterTwinPanel_phoenixContent_WorkCalendarTab_PharAdminScreensCalendarsmainUW
orkCalendarsContainterascx_0_workCalendarProfile_btnSave");
    GHP.ClickById("btnYES");
}

public void DeleteWorkCalendar(string text)
//odebrání pracovního kalendáře
{
    OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    PC.SetFilterAndOpenFirstItemFromDg("dtgWorkCalendars", "1", text, false);
    GHP.ClickById("btnDelete");
    GHP.ClickById("btnYES");
}

//(výjimky)
public void AddExcept(string text)
//vytvoření vyjimky (týdenní)
{
    OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());

    GHP.ClickByNameNowait("ctl00$MasterTwinPanel$phoenixContent$WorkCalendarItemsTabBar$T
abButtonExceptions$ctl00");

    GHP.ClickByNameNowait("ctl00$MasterTwinPanel$phoenixContent$PharAdminScreensCalendars
mainUExceptionsTabascx_1$ExceptionsTabBar$TabButtonExceptions$ctl00");
    GHP.ClickById("btnNewWorkCalendarException");
    GHP.TypeById("ebName", text);
    GHP.TypeById("ebTimeFrom1", "12:00");
    GHP.TypeById("ebTimeTo1", "23:00");
    GHP.TypeById("ebEfficiency1", "11:00");
    GHP.TypeById("ebTheNumberOfStaff1", "4");
    GHP.ClickById("rbtnListRepeat_1");
    GHP.TypeById("ebWeeklyEveryWeek", "2");

    GHP.ClickById("MasterTwinPanel_phoenixContent_PharAdminScreensCalendarsmainUException
sTabascx_1_PharAdminScreensCalendarsmainUExceptionsContainterascx_0_exceptionProfile_
chkbDay2");
    GHP.TypeById("dtpRecurrenceRangeStart_EBDate", "20.7.2017");
    GHP.ClickById("rbRangeEndChoice2");
    GHP.TypeById("dtpRecurrenceRangeEnd_EBDate", "23.11.2017");
    GHP.ClickById("btnSave");
}

```

```

        public void AddExceptMonth(string text)
        //vytvoření vyjímky (měsíční)
        {
            OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
            GHP.ClickByNameNowait("ctl00$MasterTwinPanel$phoenixContent$WorkCalendarItemsTabBar$TabButton
            Exceptions$ctl00");
            GHP.ClickByNameNowait("ctl00$MasterTwinPanel$phoenixContent$PharAdminScreensCalendarsmainUExc
            eptionsTabascx_1$ExceptionsTabBar$TabButtonExceptions$ctl00");
                GHP.ClickById("btnNewWorkCalendarException");
                GHP.TypeById("ebName", text);
                GHP.TypeById("ebTimeFrom1", "12:00");
                GHP.TypeById("ebTimeTo1", "23:00");
                GHP.TypeById("ebEfficiency1", "11:00");
                GHP.TypeById("ebTheNumberOfStaff1", "4");
                GHP.ClickById("rbtnListRepeat_2");
                GHP.ClickById("rbMonthlyChoice2");
                GHP.SelectById("ddlMonthlyNthDay", "value=2");
                GHP.SelectById("ddlMonthlyDay", "value=8");
                GHP.TypeById("ebMonthlyOfEveryMonth2", "3");
                GHP.TypeById("dtpRecurrenceRangeStart_EBDate", "20.7.2017");
                GHP.ClickById("rbRangeEndChoice1");
                GHP.TypeById("ebEndAfterFixedOccurrences", "3");
                GHP.ClickById("btnSave");
            }

        public void AddExceptYear(string text)
        //vytvoření vyjímky (roční)
        {
            OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
            GHP.ClickByNameNowait("ctl00$MasterTwinPanel$phoenixContent$WorkCalendarItemsTabBar$T
            abButtonExceptions$ctl00");
            GHP.ClickByNameNowait("ctl00$MasterTwinPanel$phoenixContent$PharAdminScreensCalendars
            mainUExceptionsTabascx_1$ExceptionsTabBar$TabButtonExceptions$ctl00");
                GHP.ClickById("btnNewWorkCalendarException");
                GHP.TypeById("ebName", text);
                GHP.TypeById("ebTimeFrom1", "12:00");
                GHP.TypeById("ebTimeTo1", "23:00");
                GHP.TypeById("ebEfficiency1", "11:00");
                GHP.TypeById("ebTheNumberOfStaff1", "4");
                GHP.ClickById("rbtnListRepeat_3");
                GHP.ClickById("rbYearlyChoice2");
                GHP.SelectById("ddlYearlyNthDay", "value=3");
                GHP.SelectById("ddlYearlyDay", "value=16");
                GHP.SelectById("ddlYearlyMonth", "value=3");
                GHP.TypeById("dtpRecurrenceRangeStart_EBDate", "20.7.2017");
                GHP.ClickById("rbRangeEndChoice1");
                GHP.TypeById("ebEndAfterFixedOccurrences", "3");
                GHP.ClickById("btnSave");
            }

        public void CheckExcept(string text)
        //kontrola vytvoření vyjímky
        {
            OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());

            GHP.ClickByNameNowait("ctl00$MasterTwinPanel$phoenixContent$WorkCalendarItemsTabBar$T
            abButtonExceptions$ctl00");

            GHP.ClickByNameNowait("ctl00$MasterTwinPanel$phoenixContent$PharAdminScreensCalendars
            mainUExceptionsTabascx_1$ExceptionsTabBar$TabButtonExceptions$ctl00");
            PC.SetFilter("dtgExceptions", "1", text);
        }

```

```

public void DelExcept(string text)
//smazání vyjímky
{
    OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());

    GHP.ClickByNameNowait("ctl00$MasterTwinPanel$phoenixContent$WorkCalendarItemsTabBar$TabButtonExceptions$ctl00");

    GHP.ClickByNameNowait("ctl00$MasterTwinPanel$phoenixContent$PharAdminScreensCalendarsmainUExceptionsTabascx_1$ExceptionsTabBar$TabButtonExceptions$ctl00");
    PC.SetFilterAndOpenFirstItemFromDg("dtgExceptions", "1", text);
    GHP.ClickById("btnDelete");
    GHP.ClickById("btnYES");
}

//Přiřazení pracovních kalendářů

public void AssignmentWorkCalendar(string text)
//přiřazení kalendáře ke stroji
{
    OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    GHP.ClickByName("TabButtonWorkCalendarAssign$ctl00");
    GHP.ClickByName("TabButtonWorkCalendarsOnEquipment$ctl00");
    PC.SetFilterAndOpenFirstItemFromDg("dtgEquipments", "1", "StrojHPP", false);
    PC.AssignWorkCalendar("Default", "TESTOVACIHP");
    Thread.Sleep(2000);
    GHP.ClickById("btnSave");
    GHP.ClickById("btnYES");
}

public void DeleteAssignmentWorkCalendar(string text)
//přiřazení kalendáře ke stroji
{
    OpenWorkCalendarsModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    GHP.ClickByName("TabButtonWorkCalendarAssign$ctl00");
    GHP.ClickByName("TabButtonWorkCalendarsOnEquipment$ctl00");
    PC.SetFilterAndOpenFirstItemFromDg("dtgEquipments", "1", "StrojHPP", false);
    GHP.ClickById("btnDelete");
    GHP.ClickById("btnYES");
}

//Zařízení

public void AddNewEquipment(string text, string longText, string number)
//Případ: Přidání nového zařízení.
{
    OpenDeviceManagementModule(System.Reflection.MethodBase.GetCurrentMethod().ToString());
    GHP.ClickById("btnAddButton");
    GHP.TypeById("ebName", text);
    GHP.TypeById("ebIdentificationNo", DateTime.Now.ToString("yyyyMMddHHmm"));
    GHP.TypeById("ebDescription", longText);
    GHP.TypeByName("ebEvNumber", "652");
    GHP.TypeByName("ebSerialNo", number);
    GHP.TypeById("ebDeviceLocation", text);
    GHP.TypeById("EBDate", DateTime.Now.ToString("dd.MM.yyyy"));
    GHP.TypeById("ebWorkingLifeCycle", number);
    GHP.TypeById("ebWorkingLifeInStandardHours", number);
    GHP.ClickByIdNowait("cbxHasMonitor");
    GHP.ClickByIdNowait("bxUseForPlanningUI");
    GHP.TypeById("ebOperationRegistrationCount", number);
    GHP.TypeById("ebAllOperationsRegistrationCount", number);
    GHP.TypeById("ebCountProducts", number);
    GHP.ClickById("btnSaveButton");
}

```